



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DISEÑO DE UNA HERRAMIENTA DE CÁLCULO PARA EVALUAR LAS PRESTACIONES DEL SISTEMA DE FRENADO DE UN VEHÍCULO FERROVIARIO

AUTOR: JOSE IGNACIO BAIXAULI GUILLOT

TUTOR: LUIS BAEZA GONZÁLEZ

Curso Académico: 2019-20

AGRADECIMIENTOS

Quiero aprovechar la ocasión para agradecer el gran apoyo recibido por parte de Iván Rubio, quien ha sido un pilar fundamental para el desarrollo de este proyecto. Iván se ha encargado en todo momento de guiarme y facilitarme todo tipo de conocimientos acerca del mundo de los vehículos ferroviarios, siendo no solamente un fantástico tutor, pero también un excelente compañero de trabajo. Además, quiero agradecer a la empresa Stadler Rail Valencia por haber confiado en mí con la beca recibida para realizar el presente trabajo de fin de grado con ellos, así como a Luis Baeza por haberme acompañado en el desarrollo de este proyecto y tutorizar y revisar el mismo. He de mencionar también a Javier Carballeira por haberme puesto en contacto con Luis.

Finalmente, quiero dedicar una especial mención a mi familia, la cual ha sido un pilar fundamental a lo largo de toda mi vida, pero especialmente quiero agradecerles su apoyo en este último curso 2019/2020 tan intenso.

RESUMEN

Debido a las implicaciones en la seguridad y las prestaciones, el cálculo y verificación de los sistemas de freno constituyen un conjunto de tareas necesarias en el proceso de diseño de los vehículos ferroviarios. La arquitectura, funcionalidad y prestaciones del sistema de freno están regulados por distintas normativas dependiendo del tipo de vehículo y su tipo de operación. La norma europea EN 14531-2 proporciona un criterio de cálculo del rendimiento de los frenos para todo tipo de composiciones de tren, unidades o vehículos aislados, incluyendo alta velocidad, las locomotoras, los coches de viajeros, el material autopropulsado y los vagones. Es importante disponer de herramientas computacionales que implementan las metodologías de la normativa, de manera que se facilite al diseñador su tarea al reducir los errores de cálculo, se posibilite un resultado óptimo y se ahorre tiempos de diseño.

En el presente trabajo de fin de grado se propone abordar el desarrollo de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario y garantizar que se cumplen los requisitos impuestos por la normativa UNE-EN 14531-2. Con el desarrollo de una herramienta que automatice estos cálculos, simplificaremos la verificación del diseño del vehículo respecto los requisitos de la normativa en función del tipo de aplicación ferroviaria (tipo de explotación y de vehículo) y se facilitará la obtención de diseños de freno óptimos.

Palabras Clave: Stadler Rail, Herramienta, Sistema de Freno, Vehículo Ferroviario.

RESUM

A causa de les implicacions en la seguretat i les prestacions, el càlcul i verificació dels sistemes de fre constitueixen un conjunt de tasques necessàries en el procés de disseny dels vehicles ferroviaris. L'arquitectura, funcionalitat i prestacions del sistema de fre estan regulats per distintes normatives depenent del tipus de vehicle i el seu tipus d'operació. La norma europea EN 14531-2 proporciona un criteri de càlcul del rendiment dels frens per a tot tipus de composicions de tren, unitats o vehicles aïllats, incloent alta velocitat, les locomotores, els cotxes de viatgers, el material autopropulsat i els vagons. És important disposar de ferramentes computacionals que implementen les metodologies de la normativa, de manera que es facilite al dissenyador la seua tasca al reduir els errors de càlcul, es possibilita un resultat òptim i s'estalvie temps de disseny.

En el present treball de fi de grau es proposa abordar el desenrotllament d'una ferramenta de càlcul per a avaluar les prestacions del sistema de frenada d'un vehicle ferroviari i garantir que es complixen els requisits imposats per la normativa UNE-EN 14531-2. Amb el desenrotllament d'una ferramenta que automatitze estos càlculs, simplificarem la verificació del disseny del vehicle respecte els requisits de la normativa en funció del tipus d'aplicació ferroviària (tipus d'exploació i de vehicle) i es facilitarà l'obtenció de dissenys de fre òptims.

Paraules clau: Stadler Rail, Ferramenta, Sistema de Fre, Vehicle Ferroviari.

ABSTRACT

Due to the implications on safety and performance, the calculation and verification of brake systems constitute a set of necessary tasks in the design process of railway vehicles. The architecture, functionality and performance of the brake system are regulated by different standards depending on the type of vehicle and its type of operation. The European standard EN 14531-2 provides a criterion for calculating brake performance for all types of train compositions, units or isolated vehicles, including high-speed, locomotives, passenger cars, self-propelled material and wagons. It is important to have computational tools that implement the methodologies of the regulations, so that the designer is facilitated in his task by reducing calculation errors, an optimal result is made possible, and design times are saved.

In this final degree project, it is proposed to address the development of a calculation tool to evaluate the performance of the braking system of a railway vehicle and ensure that the requirements imposed by the UNE-EN 14531-2 standard are met. With the development of a tool that automates these calculations, we will simplify the verification of the vehicle design with respect to the regulatory requirements depending on the type of railway application (type of operation and vehicle) and will facilitate the obtaining of optimal brake designs.

Keywords: Stadler Rail, Tool, Brake System, Rail Vehicle.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	16
1.1. JUSTIFICACIÓN.....	16
1.2. INFORMACIÓN DE LA EMPRESA	17
1.3. OBJETO DE PROYECTO	19
CAPÍTULO 2. DESCRIPCIÓN DEL PROBLEMA	20
CAPÍTULO 3. NORMATIVA UNE-EN 14531-2	22
3.1. SÍMBOLOS E ÍNDICES	22
3.2. CARACTERÍSTICAS GENERALES.....	26
3.3. CARACTERÍSTICAS DEL TIPO DE EQUIPO DE FRENO.....	26
3.4. CARACTERÍSTICAS DE PUESTA EN MARCHA Y FUNCIONAMIENTO.....	27
3.5. FUERZA DEL FRENO DE DISCO	27
3.6. OTROS PARÁMETROS DEL MODELO.....	29
CAPÍTULO 4. DISEÑO DE LA HERRAMIENTA.....	30
4.1. USUARIOS Y STAKEHOLDERS	30
4.2. REQUERIMIENTOS DE USUARIO	31
4.3. REQUISITOS DEL SISTEMA	31
4.4 DESCRIPCIÓN DE LA ESTRUCTURA DEL PROGRAMA	33
4.5. FUENTE DE DATOS	35
CAPÍTULO 5. MANUAL DE USUARIO.....	44
5.1 INSTALACIÓN DEL SOFTWARE	44

5.2 EJECUCIÓN DEL PROGRAMA	45
5.3 ANÁLISIS DE RESULTADOS	46
CAPÍTULO 6. PLAN DE DESARROLLO E IMPLEMENTACIÓN	50
CAPÍTULO 7. RESULTADO DEL PROYECTO	54
CAPÍTULO 8. CONCLUSIONES.....	58
CAPÍTULO 9. BIBLIOGRAFÍA	60
ANEXO I. PRESUPUESTO	61
I.I. CUADRO DE COSTES BÁSICOS.....	61
I.II. CUADRO DE PRECIOS DESCOMPUESTOS.....	62
I.III. PRESUPUESTO TOTAL	63
ANEXO II. CÓDIGO.....	64
SCRIPT 1. LOAD_DATA	64
SCRIPT 2. MAIN.....	65
SCRIPT 3. LOAD_DIAM.....	99
SCRIPT 4. FORCES_CALCULATION	102
SCRIPT 5. VECTORS	115
SCRIPT 6. EXPORT_COND1_TXT	123
SCRIPT 7. FILE2EXCEL.....	131

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Historia de Stadler Rail Valencia.....	17
<i>Figura 2.</i> Planta de producción Stadler Rail Valencia.....	18
<i>Figura 3.</i> Herramienta de cálculo anterior en forma de hoja Excel.....	21
<i>Figura 4.</i> Esquema de usuarios y stakeholders.....	30
<i>Figura 5.</i> Masas nominales y masas rotativas.	36
<i>Figura 6.</i> Masas dinámicas.	36
<i>Figura 7.</i> Diámetros de las ruedas.	37
<i>Figura 8.</i> Resistencia del vehículo.....	37
<i>Figura 9.</i> Características de freno de fricción.....	38
<i>Figura 10.</i> Características de freno dinámico.	39
<i>Figura 11.</i> Características de freno de pista.....	40
<i>Figura 12.</i> Condiciones degradadas.....	41
<i>Figura 13.</i> Presiones disponibles.	42
<i>Figura 14.</i> Velocidades iniciales, paso del tiempo, y coeficientes de fricción disponibles.....	43
<i>Figura 15.</i> Elementos de la carpeta de trabajo <i>STADLER</i>	45
<i>Figura 16.</i> Introducción de parámetros de entrada en MATLAB.	46
<i>Figura 17.</i> Hoja <i>Data</i> del archivo <i>Results.xlsx</i>	46
<i>Figura 18.</i> Valor de las fuerzas en función de la velocidad del vehículo.....	47
<i>Figura 19.</i> Velocidad del vehículo y distancia de frenado.	47
<i>Figura 20.</i> Distancia de frenado en función del tiempo.....	48
<i>Figura 21.</i> Tiempo de cálculo del programa.....	55

ÍNDICE DE TABLAS

<i>Tabla 1.</i> Símbolos.	24
<i>Tabla 2.</i> Índices.	25
<i>Tabla 3.</i> Información de los stakeholders del proyecto.	31
<i>Tabla 4.</i> Requisitos mínimos.	33
<i>Tabla 5.</i> Esquema de la duración del proyecto.	50
<i>Tabla 6.</i> Sueldos de convenio Stadler Rail.	56

CAPÍTULO 1. INTRODUCCIÓN

1.1. JUSTIFICACIÓN

Este trabajo final de grado ha sido desarrollado como parte de una beca en la Cátedra Stadler dentro de la Universidad Politécnica de Valencia. La empresa Stadler Rail es una empresa líder en el sector ferroviario mundial y que se ha consolidado como un referente en el mercado español tras adquirir a Vossloh la planta española en 2016, aprovechando el proceso de desinversiones de su anterior propietario.

Gran parte del éxito de los productos de Stadler Rail lo aglutina sus locomotoras, y entre ellas destaca la buena acogida en el mercado de la Eurodual, una máquina desarrollada por los ingenieros de la planta española que combina la tracción eléctrica con la tracción diésel. Su doble tecnología de tracción y el uso de tres bogies con un desgaste de la infraestructura similar al de dos convencionales dotan a esta locomotora de una mayor capacidad de arrastre que seduce a los grandes clientes. Tales factores y una exitosa política comercial explican el crecimiento que está experimentando Stadler Valencia durante los últimos años. Además, la respuesta de Stadler a un sector ferroviario en continuo desarrollo se basa en el diseño de vehículos innovadores y modulares para cada segmento de mercado, que se caracterizan por un rápido desarrollo, unos costes de operación reducidos y una flexibilidad extraordinaria.

En la actualidad, independientemente del tipo de vehículo que se fabrique, el cálculo del sistema de freno y de sus prestaciones es un requisito fundamental e indispensable en el proceso de diseño. Para evaluar las prestaciones del sistema de frenado y garantizar que se cumplen los requisitos impuestos por la normativa UNE-EN 14531-2 con título *Métodos para el cálculo de las distancias de frenado de parada y desaceleración e inmovilización*, sin exceder los límites de operación de los distintos componentes y/o subsistemas que lo componen, que será el tema principal en el que se centrará este proyecto, Stadler dispone de una herramienta de cálculo Excel.

El problema que se plantea es que su herramienta no permite testear un gran número de modos degradados, pues la hoja de cálculos Excel cuenta con un rendimiento bajo a la hora de realizar cálculos complejos.

Ya que esta es una necesidad muy concreta para Stadler, no existe una herramienta desarrollada por terceros que permita agilizar el cálculo del sistema de freno y de sus prestaciones con

distintos modos degradados. Para ello, la empresa ha realizado un convenio con la Universidad Politécnica de Valencia para poder diseñar, desarrollar e implementar una solución concreta a este problema al que se enfrenta, y es lo de que ha conseguido con este trabajo de fin de grado.

1.2. INFORMACIÓN DE LA EMPRESA

La historia de Stadler Rail Valencia S.A.U se remonta a 1897, cuando dos herreros valencianos, Miguel Devís y José Noguera Chulià, fundaron Talleres Devís en Marxalenes. Lo que inicialmente fue una calderería se ha convertido en una factoría en Albuixech de trenes, locomotoras, 'bogies' o vagones de metro con una plantilla de 900 personas que ha superado todas las crisis desde entonces. Colectivizada durante la Guerra Civil por el bando republicano, se convirtió en una de las pocas industrias pujantes en la posguerra (Romero, 2016).



Figura 1. Historia de Stadler Rail Valencia.

El afianzamiento de la empresa se produjo durante la dictadura de Primo de Rivera. En 1928, tras el fallecimiento del respetado Miguel Devís, la empresa pasó a renombrarse Construcciones Devís. Trabajadores de la fábrica tomaron el control y la pusieron al servicio de la República. Sin embargo, la derrota del régimen democrático no significó el cierre de la empresa. Al contrario, los dueños recuperaron la propiedad y en 1941 la pusieron al servicio de la recién creada Red de Ferrocarriles Españoles (Renfe). La compañía llegó a emplear 2.000 personas y se crearon talleres tanto en Alcázar de San Juan como en Carcaixent (Signes y

Capilla, 2013).

Durante esos años, se aportó capital por parte de la burguesía valenciana para apoyar el proyecto industrial. En 1947, fruto de la fusión con la catalana Sociedad Material para Ferrocarriles y Construcciones, nació Macosa, con planta en Barcelona. Tanto Renfe como General Motors pasaron a aliado tecnológico y cliente (Romero, 2016).

La llegada de la democracia y las crisis industriales de los setenta y ochenta tampoco acabaron con la fábrica, ya reubicada en Albuixech para poder hacer frente a la cartera de pedidos. En 1989 el gigante francés Alstom adquirió a Macosa. Sin embargo, Alstom vivió su propia crisis y la fábrica fue adquirida por Vossloh AG en 2005 (Romero, 2016).



Figura 2. Planta de producción Stadler Rail Valencia.

El 1 de enero de 2016 Stadler compró a Vossloh el negocio español de locomotoras. En las instalaciones de Valencia, alrededor de 900 personas trabajan en una superficie de 200.000 m² que incluye oficinas, plantas de producción, zonas de almacenaje y zonas de inspección de vehículos y ensayos.

La compañía Stadler Rail Valencia S.A.U tiene las ideas claras respecto a los valores que han de representar sus vehículos. Su objetivo es trabajar con tecnologías de estado del arte y siempre mirar al futuro para desarrollar los vehículos perfectos para sus clientes. Centrándose en la

eficiencia, gran disponibilidad, baja demanda de mantenimiento y consumo energético de sus productos. Esto les permite ofrecer a sus clientes las soluciones más económicas. El buen trato que se le da al cliente se demuestra desde la primera interacción hasta después de la entrega, tratándolos como lo más importante. Finalmente cabe destacar la flexibilidad de sus productos, ya que tanto ingenieros como diseñadores, prestando especial atención a todas las necesidades de sus clientes, logran hacer realidad todas sus peticiones.

1.3. OBJETO DE PROYECTO

El objeto del presente trabajo de fin de grado es desarrollar una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario y garantizar que se cumplen los requisitos impuestos por la normativa UNE-EN 14531-2.

Primeramente, se analizará la situación actual del cliente para definir las métricas que determinarán el éxito del proyecto. A partir de las necesidades transmitidas por el cliente, se valorará el entorno de desarrollo que mejor se ajuste a estas mismas. Una vez definido el entorno de trabajo, se procederá a desarrollar el código correspondiente para que el programa realice los cálculos necesarios para la evaluación del sistema de frenado del vehículo que se quiera diseñar.

Para el desarrollo de los cálculos, se ha de cumplir la normativa UNE-EN 14531-2, es decir, las ecuaciones de cálculo a desarrollar serán las mismas que las establecidas por la mencionada normativa. Se comenzará desarrollando el programa para el cálculo de una sola condición degradada, y una vez verificado su correcto funcionamiento, se añadirá la posibilidad de realizar los cálculos para distintas condiciones degradadas.

Posteriormente al desarrollo de los cálculos establecidos por la normativa, se ha de crear una plantilla que permita visualizar los resultados obtenidos para corroborar el correcto funcionamiento del programa.

Cuando el programa esté completo se ha de testear con el cálculo de una condición degradada para comparar el rendimiento de la herramienta creada con el rendimiento de la herramienta anterior. Se ha de evaluar tanto la velocidad de cálculo como el valor del coeficiente de desviación máximo establecido por la normativa UNE-EN 14531-2.

CAPÍTULO 2. DESCRIPCIÓN DEL PROBLEMA

En la actualidad, la empresa Stadler Rail Valencia diseña y fabrica vehículos ferroviarios destinados al transporte de pasajeros y mercancías. Independientemente del tipo de vehículo, el cálculo del sistema de freno y de sus prestaciones es un requisito fundamental e indispensable en el proceso de diseño.

La arquitectura, funcionalidad y prestaciones del sistema de freno están regulados por distintas normativas dependiendo del tipo de vehículo y su ámbito de operación.

Stadler dispone de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado y garantizar que se cumplen los requisitos impuestos por la normativa UNE-EN 14531-2 sin exceder los límites de operación de los distintos componentes y/o subsistemas que lo componen. Las prestaciones deben evaluarse en los diversos modos de operación entre los que se incluye toda la casuística de modos degradados.

Para poder realizar los cálculos establecidos por la norma UNE-EN 14531-2, con título *Métodos para el cálculo de las distancias de frenado de parada y desaceleración e inmovilización*, los responsables del departamento de frenos de Stadler utilizan una plantilla Excel que, introduciendo unos parámetros de entrada, realiza todas las iteraciones necesarias. Al ser cálculos complejos, el rendimiento de Excel supone un problema, ya que el tiempo de espera hasta mostrar los resultados es muy extenso.

Con la herramienta anterior, los usuarios tardan alrededor de 2 horas en comprobar el resultado de los cálculos, lo cual impide poder evaluar distintos modos degradados de una manera ágil y dinámica.

Así pues, con el desarrollo de una herramienta en MATLAB, se han optimizado los algoritmos, cálculos y otros cuellos de botella que facilitan la agilidad de los resultados.

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario

CityLink Dual		Vehicle mass (kg)	Wheel diam. (mm)	Speed (km/h)								
		69377	720	50								
INPUTS					RESULTS							
General data					Equivalent	3,01	m/s ²				Error	0,0090
Load condition	LOAD 1	M _{it}	69377 kg	Average dec	2,34	m/s ²				Calculation OK		
		M _{9m}	74069 kg	Stopping tir	5,3	s						
Wheel diameter	New	D	720 mm	Equivalent	0,7	s						
				Stopping di	41	m						
Time step	0,05	s		Freewheel	9,1	m						
Initial speed	50	km/h		Maximum j	6,00	m/s ³						
Friction coefficient	0,35			Total energ	7,14	MJ						
Brake systems data:												
Friction brake												
	MB1	TB2	TB3	MB4	TB5	TB6	MB7	TB8	MB9			
Bogie type	MB	TB	TB	MB	TB	TB	MB	TB	MB			
Active brakes	2	4	4	2	0	0	0	0	0			
Cylinder pressure	100	100	100	100	100	100	100	100	100	bar		
Pad Brake Force	0	0	0	0	0	0	0	0	0	N		
Braking force	0	0	0	0	0	0	0	0	0	N		
End step 1	60	80	80	60	80	80	60	80	60	km/h		
Cylinder pressure	100	42,3	42,3	100	42,3	42,3	100	42,3	100	bar		
Pad force per disc	0	28238	28238	0	28238	28238	0	28238	0	N		
Braking force	0	22292	22292	0	0	0	0	0	0	N		
Start step 2	56	75	75	56	75	75	56	75	56	km/h		
%Fmax Step 1	0%	0%	0%	0%	0%	0%	0%	0%	0%			
%Fmax Step 2	0%	42%	42%	0%	42%	42%	0%	42%	0%			
Electro-dynamic brake												
	MB1	TB2	TB3	MB4	TB5	TB6	MB7	TB8	MB9			
Bogie type	MB	TB	TB	MB	TB	TB	MB	TB	MB			
Active motors	2	0	0	2	0	0	0	0	0	4		
Application	90%	0%	0%	90%	0%	0%	0%	0%	0%			
Fmax (per bogie)	58,5	0,0	0,0	58,5	0,0	0,0	0,0	0,0	0,0	117	KN	
Magnetic brake												
	MB1	TB2	TB3	MB4	TB5	TB6	MB7	TB8	MB9	VEHICLE		
Active brakes	2	2	2	2	0	0	0	0	0	8		
Fmag (N)	14950	14950	14950	14950	0	0	0	0	0	59800		
External forces:												
		N							CALCULATION			
Vehicle resistance	ON	1318,25							CALCULATION DIAMETERS & LOAD 1			
Slope	0,0	0,0							CALCULATION DIAMETER & LOAD 2			
RESULTS												
											CALCULATION DIAMETERS & LOAD 3	
											ALL CALCULATION	
											CLEAN RESULTS TABLE	

Figura 3. Herramienta de cálculo anterior en forma de hoja Excel.

CAPÍTULO 3. NORMATIVA UNE-EN 14531-2

La norma hace referencia a aplicaciones ferroviarias, concretamente a los métodos para el cálculo de las distancias de frenado, de parada, desaceleración e inmovilización. En ella se desarrollan paso a paso los cálculos a realizar para composiciones de tren o vehículos aislados.

Esta norma es la versión oficial de la Norma Europea EN 14531-2:2015 y anula y sustituye a la Norma UNE-EN 14531-6:2010. Dicha norma europea ha sido aprobada por el Comité Europeo de Normalización el 27-06-2015.

Esta norma europea describe un método de cálculo común para aplicaciones ferroviarias. Se describen los algoritmos generales que utilizan el cálculo paso a paso para su uso en el diseño, la validación de los equipos de freno y su rendimiento en todo tipo de composiciones de tren y vehículos aislados. Además, permite la verificación mediante cálculo del rendimiento de la parada y desaceleración para trenes de alta velocidad y convencionales, que se explotan en la infraestructura de alta velocidad y convencional.

El método propuesto de esta norma se basa en un algoritmo numérico de integración de tiempo. La norma explica un esquema de integración numérica sencillo con el fin de proporcionar un ejemplo claro y útil del método propuesto.

3.1. SÍMBOLOS E ÍNDICES

Para la finalidad de este documento, se aplican los símbolos e índices dados en las tablas 1 y 2 respectivamente.

Símbolo	Definición	Unidad
A	Área	m ²
a	Desaceleración	m/s ²
D	Diámetro de rueda	m
F	Fuerza	N
F _B	Fuerza de frenado	N
ε	Desviación aceptable de v_{fin} utilizada para detener el cálculo del paso del tiempo	m/s
g _n	Aceleración normal en caída libre = 9,086 m/s ²	m/s ²
i	Pendiente	-
i _c	Cilindro/coeficiente de la unidad	-
i _{rig}	Coeficiente de aparejo	-
i _{tra}	Coeficiente de transmisión	-
<i>m</i>	Masa	kg
<i>n</i>	Cantidad	-
<i>p</i>	Presión	Pa
P	Potencia del equipo de freno	W
<i>r</i>	Radio	m

s	Distancia	m
t	Tiempo	s
W_b	Energía disipada por el equipo de freno	J
τ	Coefficiente de adherencia	-
μ	Coefficiente de fricción	-
η	Eficiencia	-

Tabla 1. Símbolos.

Índice	Definición
a_x	Eje
a	Disponible
B	Freno/frenado
BED	Fuerza de freno electrodinámico
C	Cilindro/unidad
dyn	Dinámico
fin	Final
e	Equivalente
ext	Externo

i	Tipo de equipo de freno
j	Paso de tiempo
máx.	Máximo
N	Nominal
Ra	Resistencia del tren al movimiento
req	Requerido
rig	Aparejo
rot	Giratorio
st	Estático
tra	Transmisión

Tabla 2. Índices.

El cálculo se presenta en un diagrama de flujo como se muestra en la figura A.1.

El algoritmo utiliza valores instantáneos que se han calculado paso a paso. La integración numérica se debe basar en el tiempo.

La precisión de los cálculos aquí descritos depende directamente de la precisión de los datos de entrada.

La precisión de los valores de los datos de entrada debe ser relevante a los efectos del cálculo, y debe ser trazable en cuanto a cómo se establecieron.

Para el cálculo de las distancias de frenado y desaceleración, se han dividido las características en características generales, características del tipo de equipo de freno, y características de puesta en marcha y funcionamiento.

3.2. CARACTERÍSTICAS GENERALES

Engloba tanto la composición del tren, es decir, cantidad de ejes, como su diámetro y masas (rotatoria, estática y dinámica).

También se incluye en esta categoría la resistencia del tren, expresada con la siguiente fórmula:

$$F_{Ra} = A + B \cdot v + C \cdot v^2$$

F_{Ra} – valor instantáneo de la resistencia del tren, en Newtons.

v – velocidad instantánea del vehículo, en m/s.

A – coeficiente característico del tren independiente de la velocidad, en Newtons.

B – coeficiente característico del tren proporcional a la velocidad, en Newtons/(m/s).

C – coeficiente característico de la resistencia aerodinámica del tren, en Newtons/(m/s)².

3.3. CARACTERÍSTICAS DEL TIPO DE EQUIPO DE FRENO

Engloba la fuerza de frenado que genera cada tipo de equipo de freno. En este trabajo nos hemos centrado concretamente en el freno de disco.

Dicha categoría contiene los datos relevantes al cálculo de la fuerza de fricción, fuerza dinámica, y fuerza de pista, todos ellos los analizaremos en profundidad más adelante en la explicación de los datos de entrada.

Tanto para el cálculo de la fuerza de fricción como para el cálculo de las fuerzas dinámica y de pista se tendrá en cuenta el concepto de Tiempo de Respuesta.

El tiempo de respuesta (t_r) de cada tipo de equipamiento de freno está compuesto por el tiempo de retardo (t_a) y el tiempo de acumulación (t_{ab}).

El tiempo de retardo (t_a) es el tiempo necesario para alcanzar un a% de la presión del cilindro.

El tiempo de acumulación (t_{ab}) es el tiempo posterior al tiempo de retardo que se necesita para alcanzar un b% de la presión del cilindro.

El tiempo de respuesta (t_r) es el tiempo necesario para obtener la presión demandada y puede calcularse utilizando la siguiente ecuación: $t_r = t_a + t_{ab}$

3.4. CARACTERÍSTICAS DE PUESTA EN MARCHA Y FUNCIONAMIENTO

Por lo general, al realizar los cálculos se asumen las condiciones de una vía recta y nivelada. Sin embargo, si se tiene en cuenta el frenado por una pendiente, el efecto de dicha pendiente quedará reflejado de la siguiente manera:

$$F_g = M_{st} \cdot g \cdot i / 1000$$

F_g – fuerza sobre el tren cuesta abajo, en kilonewtons (kN);

M_{st} – masa estática del tren, en toneladas (t);

g – aceleración debida a la gravedad (9,81 m/s²);

i – pendiente, en mm/m.

Se tendrá en cuenta a su vez tanto la velocidad inicial v_0 como el coeficiente de adherencia disponible. Si la adherencia requerida supera la adherencia disponible, puede conllevar un incremento de la distancia de frenado si se compara con el cálculo teórico.

Además, los cálculos se deberán realizar con todos los tipos de equipo de freno en buen estado y con una cantidad especificada.

Como último paso antes de realizar los cálculos, se considerarán situaciones degradadas que puedan afectar el rendimiento de los frenos. Entraremos más en detalle al analizar los datos de entrada del programa.

3.5. FUERZA DEL FRENO DE DISCO

Para obtener la fuerza del freno de disco comenzamos calculando la fuerza producida por el cilindro, F_c , en Newtons. Lo haremos mediante la siguiente fórmula,

$$F_c = (10 \cdot p_c \cdot A_c \cdot \eta_c) - F_{R1} - F_{R2}$$

P_c – presión en el cilindro de freno, en bar;

A_c – área del pistón de freno, en cm²;

η_c – rendimiento del cilindro.

F_{R1} – fuerza del muelle de llamada, en Newtons;

F_{R2} – fuerza opuesta al regulador, en Newtons.

Continuamos calculando la fuerza de aplicación en el ferodo, F_b , en Newtons. La fórmula será la siguiente,

$$F_b = (F_c \cdot i_c \cdot \eta_{dyn}) / n_{b,c}$$

F_c – fuerza producida por el cilindro, en Newtons;

i_c – ratio total de timonería después de cilindro;

η_{dyn} – rendimiento de timonería en condiciones dinámicas;

$n_{b,c}$ – número de ferodos por cilindro.

Seguidamente calculamos la fuerza de frenado por eje, F_{ba} , en Newtons, de la siguiente manera.

$$F_{ba} = F_b \cdot r_s / (D/2) \cdot n_{b,c} \cdot n_{c,a}$$

F_b – es la fuerza de aplicación en el ferodo, en Newtons; ^{[1][2]}_[SEP]

μ – es el coeficiente de fricción del ferodo;

r_s – es el radio total de barrido medio, en metros;

D – es el diámetro de la rueda, en metros;

$n_{b,c}$ – es el número de ferodos por cilindro;

$n_{c,a}$ – es el número de cilindros por eje.

Una vez tenemos todas las anteriores fuerzas calculadas, para poder obtener la desaceleración, sumaremos las fuerzas exteriores F_{RA} y F_g a la fuerza de frenado por eje F_{ba} , lo que nos dará el valor de la fuerza total, $F_T = F_{RA} + F_g + F_{ba}$.

Así, la desaceleración del vehículo vendrá dada por la siguiente expresión:

$$a_i = F_T / M_{dyn}$$

F_T – es la fuerza de frenado, en kilonewtons, [SEP]

M_{dyn} – es la masa dinámica, ($= M_{st} + M_{rot}$), en toneladas.

3.6. OTROS PARÁMETROS DEL MODELO

La velocidad en el paso t_{j+1} viene dada por la siguiente fórmula:

$$v_{j+1} = v_j - a_j \cdot \Delta t$$

La distancia en el paso t_{j+1} viene dada por la siguiente fórmula:

$$x_{j+1} = x_j + v_j \cdot \Delta t - \frac{1}{2} \cdot a_j \cdot \Delta t^2$$

La desaceleración en el paso t_{j+1} viene dada por la siguiente fórmula:

$$a_{j+1} = (F_T)_{j+1} / M_{dyn}$$

El siguiente tiempo de paso queda expresado de la siguiente manera:

$$t_{j+1} = t_j + \Delta t$$

El incremento del tiempo de paso queda expresado de la siguiente manera:

$$j = j + 1$$

a_j – es la desaceleración del vehículo en el tiempo t_j , en m/s^2 ;

F_T – es la fuerza total en el tiempo t_j , en kN;

j – es el número del paso de integración;

M_{dyn} – es la masa dinámica, ($= M_{st} + M_{rot}$), en toneladas;

x_j – es la distancia en el tiempo t_j , en m/s;

Δt – es el paso de tiempo, en s;

t_j – es el tiempo de desaceleración en el paso de integración j , en s.

CAPÍTULO 4. DISEÑO DE LA HERRAMIENTA

4.1. USUARIOS Y STAKEHOLDERS

Las partes interesadas, más comúnmente conocidas como stakeholders, son parte fundamental de la totalidad de nuestro proceso de desarrollo y diseño de experiencia de usuario.

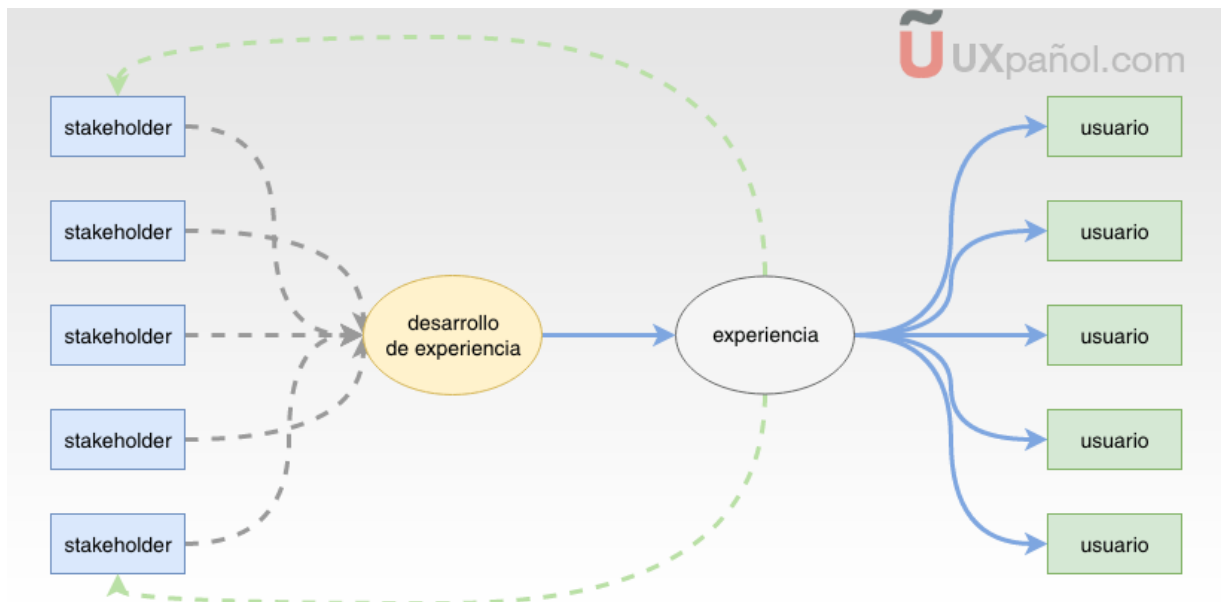


Figura 4. Esquema de usuarios y stakeholders.

Como bien podemos apreciar en la imagen, los stakeholders son una pieza clave en el desarrollo de la herramienta o experiencia, pudiendo intervenir directa o indirectamente (líneas punteadas). Finalizado el desarrollo de la herramienta, son los usuarios los que hacen uso de ella. Los stakeholders también pueden ser usuarios finales, como en este proyecto.

En este trabajo de fin de grado se ha contado con tan solo un stakeholder, el cual hará también de usuario final.

Nombre	Descripción	Responsabilidades
Iván Rubio	Ingeniero Mecánico	Representa a los usuarios posibles del sistema y aprueba los requerimientos, alcance y funcionalidades de la herramienta.

Tabla 3. Información de los stakeholders del proyecto.

4.2. REQUERIMIENTOS DE USUARIO

El sistema deberá permitir a los usuarios introducir como archivo de entrada cualquier documento Excel independientemente de su ubicación en el ordenador. Además, permitirá introducir como variables de entrada tanto las velocidades iniciales como los modos degradados que se quieran calcular. Una vez el usuario haya introducido estos parámetros de entrada y se haya ejecutado el programa, el sistema permitirá guardar los resultados en archivos .txt en función de la velocidad inicial, diámetro de las ruedas y carga del vehículo. De esta manera, nos será más fácil controlar los resultados, y además nos permitirá exportarlos de forma individual a una plantilla Excel para comprobar gráficamente que efectivamente no ha habido ningún error en el cálculo.

4.3. REQUISITOS DEL SISTEMA

Para poder hacer uso de la herramienta, el usuario final tendrá que tener acceso a MATLAB, un lenguaje de cálculo técnico que ofrece un potente entorno de programación para desarrollo de algoritmos, análisis de datos, visualización y cálculo numérico.

La plataforma de MATLAB está optimizada para resolver problemas científicos y de ingeniería. El lenguaje de MATLAB, basado en matrices, es la forma más natural del mundo para expresar las matemáticas computacionales. Las gráficas integradas facilitan la visualización de los datos y la obtención de información a partir de ellos.

Una vasta biblioteca de herramientas (Toolboxes) integradas le permite empezar a trabajar inmediatamente con algoritmos esenciales para su dominio. El entorno de escritorio invita a experimentar, explorar y descubrir. Todas estas herramientas y funciones de MATLAB están probadas rigurosamente y diseñadas para trabajar juntas.

Sus características principales son las siguientes:

- Lenguaje de alto nivel para cálculos científicos y de ingeniería.
- Entorno de escritorio optimizado para la exploración iterativa, el diseño y la solución de problemas.
- Gráficas para visualizar datos y herramientas para crear diagramas personalizados.
- Aplicaciones para ajustar curvas, clasificar datos, analizar señales, ajustar sistemas de control y muchas otras tareas.
- Toolboxes complementarias para una amplia variedad de aplicaciones científicas y de ingeniería.
- Herramientas para crear aplicaciones con interfaces de usuario personalizadas.
- Interfaces para C/C++, Java®, .NET, Python, SQL, Hadoop y Microsoft® Excel®.
- Opciones de implementación libres de derechos para compartir programas de MATLAB con los usuarios finales.

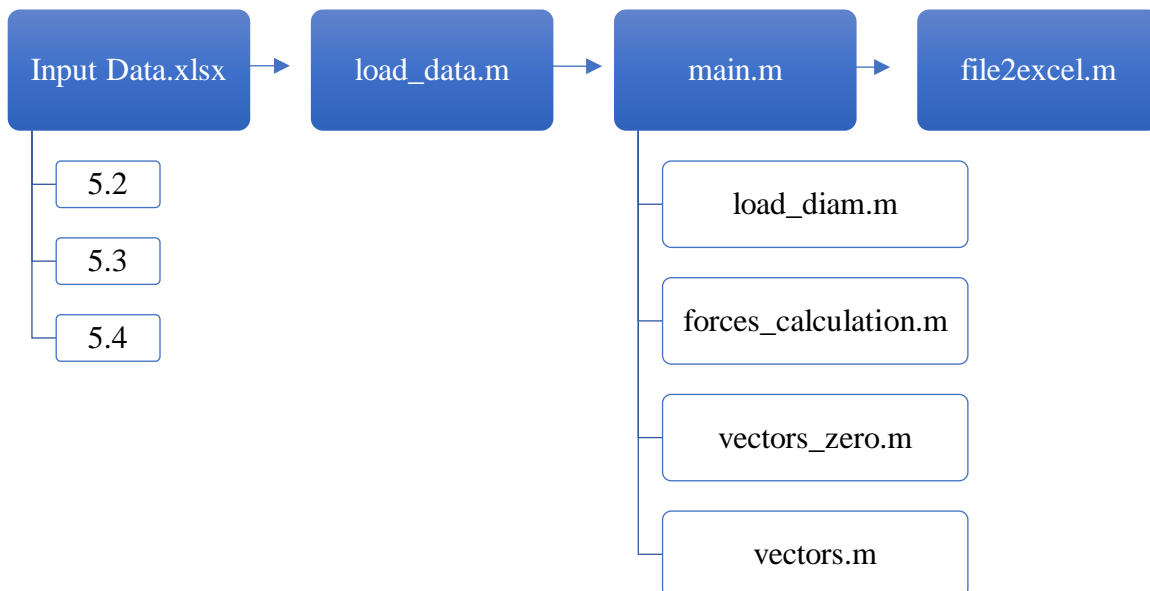
Para poder descargarse el software MATLAB los usuarios deben contar con los siguientes requisitos mínimos.

	Requisito mínimo
Matlab	R2016b en adelante (se recomienda R2018b)
Procesador	Procesador Intel o AMD x86-64 con soporte de instrucciones AVX2
Sistema operativo	Windows: Windows 7 SP1 en adelante, Windows Server 2008 SP2 en adelante Mac: macOS 10.10 - 10.11 en adelante Linux: Kernel 2.6 o superior, glibc 2.11 o superior

RAM	1 GB o más (se recomiendan 4 GB)
Tarjeta gráfica	Soporte para OpenGL 3.3 recomendado con 1GB en GPU
Internet	Es preciso disponer de conexión a Internet
Licencia	Se necesita una licencia para poder disponer de todas las funcionalidades de la herramienta Matlab

Tabla 4. Requisitos mínimos.

4.4 DESCRIPCIÓN DE LA ESTRUCTURA DEL PROGRAMA



El programa comienza leyendo un archivo Excel, *Input Data*, con toda la información relevante del vehículo ferroviario que se desea diseñar. El archivo cuenta con 3 hojas principales, 5.2, 5.3

y 5.4, las cuales se han explicado en detalle en el siguiente apartado “Fuente de datos”.

El script de MATLAB *load_data* lee la información del archivo de entrada y la guarda en distintas tablas en función del tipo de característica del vehículo. Estas tablas son las siguientes:

NominalMass_Data, almacena las masas nominales de los distintos vagones.

DynamicMass_Data, almacena las masas dinámicas de los distintos vagones.

WheelDiameter_Data, almacena los distintos diámetros de rueda disponibles.

VehicleResistance_Data, almacena las variables de resistencia de cada vagón.

FrictionBrakeForces_Data, almacena la información relevante al freno de fricción.

DynamicBrakeForces_Data, almacena la información relevante al freno dinámico.

TrackBrake_Data, almacena la información relevante al freno de pista.

Initial_Data, almacena los coeficientes de fricción disponibles y el error de cálculo admisible.

Degraded_Data_1, almacena las condiciones degradadas.

Degraded_Data_2, almacena más condiciones degradadas.

Degraded_Data_3, almacena más condiciones degradadas.

Degraded_Data_4, almacena más condiciones degradadas.

Degraded_Data_5, almacena más condiciones degradadas.

Pressure_Vel_Data, almacena las presiones del freno en función de la velocidad del vehículo.

Una vez se ha ejecutado el script *load_data* se inicia el script *main*, el cuerpo del programa. Dicho script realiza todos los cálculos necesarios siguiendo la norma UNE-EN 14531-2 llamando a distintas funciones.

Inicialmente, al ejecutar *main* el programa solicita al usuario introducir la velocidad inicial de cálculo, es decir, la velocidad a partir de la cual comenzará a frenar el vehículo ferroviario. Además, el usuario ha de introducir manualmente el valor del paso del tiempo para cada iteración, así como la cantidad de condiciones degradadas que se quiera calcular.

Una vez introducidos manualmente los parámetros de entrada, el programa llama a la función *load_diam*, la cual comprueba las veces que tiene que llamar a la función *forces_calculation* dependiendo de los datos de entrada que se hayan especificado para la carga y el diámetro. Para ello, la función *load_diam* comienza inicializando a cero los vectores que almacenan las velocidades, tiempos, fuerzas, distancias, y aceleraciones para cada iteración. *vectors_zero* es el script que realiza dicha acción. Seguidamente, en función de la carga y el diámetro que se quieran calcular – valores previamente definidos en el archivo de entrada para la condición

degradada a calcular – el programa ejecuta el script *forces_calculation* un número de veces u otro. Como existen tres tipos de diámetro (*New*, *Half* y *Worn*) y cinco cargas posibles (*Load1*, *Load2*, *Load3*, *Load4* y *Load5*), existen hasta 15 combinaciones distintas, es decir, se podrá ejecutar hasta 15 veces el script *forces_calculation*.

Seguidamente, el script *forces_calculation* realiza todos los cálculos establecidos en la norma UNE-EN 14531-2. Para calcular la deceleración que producirán los frenos, el programa comienza calculando la fuerza de frenado de cada *bogie* o vagón. Una vez calculadas las distintas fuerzas de frenado se realiza el sumatorio para calcular la total. Con la fuerza de frenado total se obtiene la deceleración, y una vez conocida la deceleración se calcula la velocidad. El script *forces_calculation* sigue iterando este proceso hasta que la velocidad calculada esté por debajo del error establecido o sea inferior a cero. Finalizados los cálculos, se ejecuta automáticamente el script *vectors*, el cual guarda los cálculos en sus respectivos vectores – en función del tipo de diámetro y tipo de carga – para después exportarlos a archivos .txt que facilitarán la visualización en Excel de los cálculos realizados.

Finalmente, para poder graficar los resultados y comprobar que no ha habido ninguna anomalía en los cálculos realizados, ejecutamos el script *file2excel* que exportará el archivo .txt a una hoja Excel – previamente diseñada para visualizar mediante gráficas los resultados.

4.5. FUENTE DE DATOS

Los datos iniciales del vehículo que se quiere diseñar quedan recogidos en un archivo de Excel con nombre *Input_Data*. Dicho archivo contiene tres hojas, las cuales han sido analizadas a continuación.

En primer lugar, la hoja 5.2 reúne los datos correspondientes a las características generales del vehículo. Dichos datos son, como bien se menciona en la norma UNE-EN 14531-2, masas nominales, masas rotativas, masas dinámicas, diámetro de la rueda, y resistencia del vehículo.

Existen distintas masas nominales disponibles (*LOAD1*, *LOAD2*, *LOAD3*, *LOAD4*, y *LOAD5*) para cada *bogie* o vagón. El programa permite calcular composiciones de trenes de hasta nueve *bogies* o vagones. Además, está la posibilidad de seleccionar *ALL* como masa nominal. En este caso, se realizará el cálculo para cada una de las cinco cargas posibles, es decir, se ejecutará el script *forces_calculation* cinco veces.

Respecto a las masas rotativas, éstas se mantienen constantes para cada estado de carga, pero sí que varían en función del *bogie* o vagón.

		Generalities: Vehicles Characteristics										
		Nominal Mass										VEHICLE
$m_{n,j}$	kg	LOAD 1	17,2	14,9	14,9	17	0	0	0	0	0	64
		LOAD 2	20,4	19,9	19,9	20,2	0	0	0	0	0	80,4
		LOAD 3	21,3	21,4	21,4	21,2	0	0	0	0	0	85,3
		LOAD 4	0	0	0	0	0	0	0	0	0	0
		LOAD 5	0	0	0	0	0	0	0	0	0	0
		ALL										
m _{jrot}	kg	Rotating Mass, m _{jrot}	1,56	0,75	0,75	1,56	5,8	5,8	5,8	5,8	5,8	33,62

Figura 5. Masas nominales y masas rotativas.

Las masas dinámicas son el resultado de la suma de las masas nominales y rotativas. De esta manera, podemos contar hasta con cinco estados de cargas dinámicas distintos para cada *bogie* o vagón.

		Dynamicl Mass									
$m_{st,j}$	kg	LOAD 1	18,76	15,65	15,65	18,56	5,8	5,8	5,8	5,8	5,8
		LOAD 2	21,96	20,65	20,65	21,76	5,8	5,8	5,8	5,8	5,8
		LOAD 3	22,86	22,15	22,15	22,76	5,8	5,8	5,8	5,8	5,8
		LOAD 4	1,56	0,75	0,75	1,56	5,8	5,8	5,8	5,8	5,8
		LOAD 5	1,56	0,75	0,75	1,56	5,8	5,8	5,8	5,8	5,8

Figura 6. Masas dinámicas.

El diámetro de las ruedas también va a definir cuántos cálculos realizará el programa. Existen tres tipos de diámetros disponibles – *New*, *Half* y *Worn*. A su vez, también es posible seleccionar *ALL* para calcular los tres tipos de diámetro, ejecutando de esta manera tres veces el script *forces_calculation*.

A diferencia de los distintos estados de carga, el valor del tipo de diámetro no varía para cada *bogie* o vagón.

	m	Wheel diameter									
		New	0,72	0,72	0,72	0,72	0,72	0,72	0,72	0,72	0,72
		Half	0,68	0,68	0,68	0,68	0,68	0,68	0,68	0,68	0,68
		Worn	0,64	0,64	0,64	0,64	0,64	0,64	0,64	0,64	0,64
		ALL									

Figura 7. Diámetros de las ruedas.

La resistencia a la rodadura es la fuerza de frenado proporcionada por la estructura del tren. Siguiendo la norma UNE-EN 14531-2, no se ha considerado el efecto de las fuerzas del viento. La siguiente tabla recoge los términos con los que se calculará la resistencia a la rodadura. Un término independiente de la velocidad del vehículo (*A*), un término proporcional a la velocidad (*B*), y un tercer término proporcional al cuadrado de la velocidad (*C*). Éstos pueden variar para cada *bogie* o vagón.

Vehicle Resistance									
$F_{Ra} = D \cdot v^3 + C \cdot v^2 + B \cdot v + A$	D1	D2	D3	D4	D5	D6	D7	D8	D9
A	561	561	561	561	0	0	0	0	0
B	19,84	19,84	19,84	19,84	0	0	0	0	0
C	0,02075	0,02075	0,02075	0,02075	0	0	0	0	0
D									

Figura 8. Resistencia del vehículo.

Seguidamente, la hoja 5.3 reúne los datos correspondientes a las características del sistema de frenado. Dichos datos son, como bien se menciona en la norma UNE-EN 14531-2, características de freno de fricción, características de freno dinámico, y características de freno de pista.

La tabla cuenta con cinco columnas que representan cinco tipos de configuración de frenos de fricción, *FBT1*, *FBT2*, *FBT3*, *FBT4* y *FBT5*. No todos los cálculos se han tenido en cuenta a la hora de realizar los cálculos, siendo los más relevantes los siguientes,

Piston Effective Section, sección efectiva del pistón, en cm².

Rigging Efficiency, eficiencia del aparejo.

Call Spring Force, fuerza del resorte, en Newtons.

Opposite Regulator Force, fuerza regulatoria contraria, en N.

Friction Radius, radio de fricción de las ruedas, en m.

Brake Disc Diameter, diámetro de los frenos de disco, en m

Rigging Ratio, ratio de aparejo.

Cylinder Efficiency, eficiencia del cilindro.

Pad Area, area de la pastilla de freno, en cm².

Number of pads per disc, número de pastillas por disco.

Además, la tabla contiene los tiempos de respuesta del freno de fricción. El tiempo de respuesta equivalente (t_{eq}), tiempo de respuesta teórico utilizado para calcular distancias de frenado, viene dado por la suma del tiempo de retardo (t_a) y la mitad del tiempo de acumulación (t_{ab}).

5.3.2	Friction Brake Forces Disc						
	Brake Type		FBT1	FBT2	FBT3	FBT4	FBT5
	Component Designation						
A_c	Piston Effective Section	[cm ²]	62	70,9	0	0	0
η_{eff}	Rigging Efficiency	[]	1	0,95	0	0	0
F_{s1}	Call Spring Force	[N]	-40000	3500	0	0	0
F_{s2}	Opposite Regulator Force	[N]	0	0	0	0	0
r_s	Friction Radius	[m]	0,195	0,194	0	0	0
D_c	Brake Disc Diameter	[m]			0	0	0
I_{rig}	Rigging Ratio	[]	2	2	0	0	0
$\eta_{cylinder}$	Cylinder Efficiency	[]	-1	1	0	0	0
$A_{s,p}$	Pad Area	[cm ²]	130	130			
$n_{b,c}$	Number of pads per disc	[]	2	2			
n	Brake Units per Calculation Division	[]					
$A_{s,c}$	Disc Area	[cm ²]	0	0	0	0	0
$n_{b,s}$	Number of disc brake faces	[]					
F_{park}	Parking Brake Force	[N]					
F_{hold}	Holding Brake Force	[N]					
F/P	Force/Pressure Ratio	[N/bar]	# REF	# REF	# REF	# REF	# REF
η_{park}	Parking/Holding Brake Units per Calc. Division	[]					
P_{rel}	Pressure Release	[bar]					
t_a	Delay Time	[s]	0,4	0,4	0	0	0
t_{ab}	Accumulation Time	[s]	1,1	1,1	0	0	0
t_{eq}	Response Equivalent Time	[s]	0,95	0,95	0,00	0,00	0,00

Figura 9. Características de freno de fricción.

La siguiente tabla recoge los cinco tipos de freno dinámico disponibles, *DBT1*, *DBT2*, *DBT3*, *DBT4*, y *DBT5*. Cada uno de ellos aplicará una fuerza de frenado dinámico en función de la velocidad a la que se desplace el vehículo.

Si el vehículo se desplaza a una velocidad inferior a v_1 ,

$$F_{dyn} = F$$

Si el vehículo se desplaza a una velocidad superior a v_1 e inferior a v_2 ,

$$F_{dyn} = F \cdot v_2 / v$$

Si el vehículo se desplaza a una velocidad superior a v_2 ,

$$F_{dyn} = F \cdot v_2 \cdot v_1 / v$$

El valor de la fuerza F y de las velocidades v_1 y v_2 variará para cada uno de los cinco modos de freno dinámicos disponibles. Además, la tabla contiene los tiempos de respuesta del freno dinámico. Éstos son, el tiempo de respuesta equivalente (t_{eq}), el tiempo de retardo (t_a) y el tiempo de acumulación (t_{ab}).

5.3.3	Dynamic Brake Forces Disc						
Brake Type			DBT1	DBT2	DBT3	DBT4	DBT5
Component Designation							
F	N		24000	32500	0	0	0
V1	km/h		100	100	100	100	100
V2	km/h		67	55	44,5	44,5	44,5
V3	km/h		0	0	0	0	0
V4	km/h		0	0	0	0	0
V5	km/h		0	0	0	0	0
Delay Time	[s]		0,2	0,2	0,2	0,2	0,2
Accumulation Time	[s]		1,1	0,3	1,1	1,1	1,1
Response Equivalent Time	[s]		0,75	0,35	0,75	0,75	0,75

Figura 10. Características de freno dinámico.

En cuanto al freno de pista, también se disponen de cinco opciones diferentes, $TBT1$, $TBT2$, $TBT3$, $TBT4$ y $TBT5$. La fuerza de frenado variará en función de la velocidad a la que se desplace el vehículo, y será distinta para cada una de las cinco opciones de freno de pista disponibles. Cada una de las filas de la tabla contiene la fuerza de frenado de cada uno de los cinco modos que actuará cuando el vehículo se desplace a la velocidad indicada, y esa fuerza se mantendrá constante hasta que el vehículo alcance la velocidad indicada en la fila inferior.

De esta manera, si el vehículo se desplaza a una velocidad inferior a 100 km/h pero superior a 89 km/h (velocidades indicadas en la primera columna), la fuerza de freno de pista será la definida para 100 km/h.

La fórmula que representa la fuerza de freno de pista es la siguiente,

$$F_{brake} = F$$

Siendo F la fuerza definida en la tabla.

5.3.4	Track Brake Disc					
	Brake Type	TBT1	TBT2	TBT3	TBT4	TBT5
	Speed (km/h)	F (N)	F (N)	F (N)	F (N)	F (N)
	100	13150	6575	0	0	0
	80	14600	6575	0	0	0
	70	15475	6575	0	0	0
	50	17525	6575	0	0	0
	40	18875	6575	0	0	0
	30	20225	6575	0	0	0
	20	20225	6575	0	0	0
	0	20225	6575	0	0	0
Delay Time	[s]	0,3	0,3	0,2	0,2	0,2
Accumulation Time	[s]	0,3	0,3	1,1	1,1	1,1
Response Equivalent Time	[s]	0,45	0,45	0,75	0,75	0,75

Figura 11. Características de freno de pista.

Finalmente, la hoja 5.4 recoge las condiciones de funcionamiento, es decir, los datos correspondientes a las condiciones degradadas de funcionamiento, las diferentes presiones disponibles, las velocidades iniciales, el paso del tiempo, y los diferentes coeficientes de fricción disponibles.

La hoja cuenta con cinco condiciones degradadas distintas, y el usuario deberá definir al ejecutar el programa qué condición o condiciones quiere calcular, pudiendo calcular todas ellas. Cada una de las condiciones es editable por el usuario, lo cual ofrece una gran variedad de casuísticas a calcular. Así pues, el ingeniero en cuestión que vaya a utilizar la herramienta podrá definir en primer lugar las distintas condiciones degradadas del vehículo en esta hoja, para después calcular el comportamiento del vehículo y comprobar si cumple la norma UNE-EN 14531-2.

De esta manera, el usuario puede editar para cada condición de funcionamiento el tipo de carga (*LOAD1, LOAD2, LOAD3, LOAD4, LOAD5 o ALL*), el tipo de diámetro (*New, Half, Worn o ALL*), el tipo de freno de fricción (*FBT1, FBT2, FBT3, FBT4 o FBT5*), el número de frenos de fricción disponibles, el coeficiente de fricción, el tipo de presión de frenado (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S o T), el tipo de freno dinámico (*DBT1, DBT2, DBT3,*

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario

DBT4 o DBT5), el número de frenos dinámicos disponibles, el porcentaje de funcionamiento del freno dinámico, el tipo de freno de pista (*TBT1, TBT2, TBT3, TBT4 o TBT5*), el número de frenos de pista, y la pendiente de la superficie por donde circula el vehículo. El tipo de carga y de diámetro es invariable para cada 'bogie' o vagón, pero el resto de elementos sí que pueden variar.

Conditions										
Designation	D1	D2	D3	D4	D5	D6	D7	D8	D9	VEHICLE
Load	LOAD 1	LOAD 1	LOAD 1	LOAD 1	LOAD 1	LOAD 1	LOAD 1	LOAD 1	LOAD 1	0
Static Mass	17,2	17,2	17,2	17,2	17,2	17,2	17,2	17,2	17,2	
Dynamic Mass	18,76	18,76	18,76	18,76	18,76	18,76	18,76	18,76	18,76	
Wheel Diameter	New	New	New	New	New	New	New	New	New	
Diameter	0,72	0,72	0,72	0,72	0,72	0,72	0,72	0,72	0,72	
Friction Type of Brake	FBT1	FBT2	FBT2	FBT1	FBT5	FBT1	FBT1	FBT1	FBT2	
Available Friction Brakes	2	2	2	2	0	0	0	0	0	8
Available Parking Brakes	2	4	4	2	0	0	0	0	0	12
Adhesion Coefficient	Friction Coeff 1	Friction Coeff 1	Friction Coeff 1	Friction Coeff 1	Friction Coeff 5	Friction Coeff 1	Friction Coeff 2	Friction Coeff 4	Friction Coeff 2	0
Friction Pressure	A	B	B	A	E	C	B	C	D	0
Dynamic Type of Brake	DBT1	DBT1	DBT1	DBT1	DBT5	DBT4	DBT3	DBT3	DBT4	
Available Dynamic Brake	2	0	0	2	0	0	0	0	0	4
Dynamic % Dynamic Brake	0,85	0	0	0,85	0	0	0	0	0	1,7
Track Type of Brake	TBT2	TBT2	TBT2	TBT2	TBT5	TBT1	TBT2	TBT3	TBT4	
Track Available Track Brake	0	0	0	0	0	0	0	0	0	0
External Force (F_{ext})										
Slope [mm/m]	0	0	0	0	0	0	0	0	0	

Figura 12. Condiciones degradadas.

La siguiente tabla recoge las distintas presiones disponibles para el cálculo de la fuerza del freno de disco. Dicha presión varía en función de la velocidad a la que se desplaza el vehículo. Se han definido unas velocidades comunes, 70, 40, 20 y 0, siendo Velocity1, Velocity2, Velocity3 y Velocity4 respectivamente. Así, si el vehículo se desplaza a una velocidad superior a Velocity1, el valor de la presión será Pressure1, y así sucesivamente hasta llegar a Velocity4.

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario

	Pressure 1	Velocity 1	Pressure 2	Velocity 2	Pressure 3	Velocity 3	Pressure 4	Velocity 4
A	100	70	100	40	100	20	100	0
B	0	70	21,4	40	21,4	20	21,4	0
C	33,4	70	33,4	40	33,4	20	33,4	0
D	2,65	70	2,65	40	2,65	20	2,65	0
E	2,65	70	2,65	40	2,65	20	2,65	0
F	2,65	70	2,65	40	2,65	20	2,65	0
G	2,65	70	2,65	40	2,65	20	2,65	0
H	2,65	70	2,65	40	2,65	20	2,65	0
I	2,65	70	2,65	40	2,65	20	2,65	0
J	2,65	70	2,65	40	2,65	20	2,65	0
K	2,65	70	2,65	40	2,65	20	2,65	0
L	2,65	70	2,65	40	2,65	20	2,65	0
M	2,65	70	2,65	40	2,65	20	2,65	0
N	2,65	70	2,65	40	2,65	20	2,65	0
O	2,65	70	2,65	40	2,65	20	2,65	0
P	2,65	70	2,65	40	2,65	20	2,65	0
Q	2,65	70	2,65	40	2,65	20	2,65	0
R	2,65	70	2,65	40	2,65	20	2,65	0
S	2,65	70	2,65	40	2,65	20	2,65	0
T	2,65	70	2,65	40	2,65	20	2,65	0

Figura 13. Presiones disponibles.

Por último, la siguiente tabla recoge el error admisible del programa, las posibles velocidades iniciales de cálculo, los distintos pasos de tiempo, y los coeficientes de fricción disponibles.

El paso del tiempo dependerá de la velocidad inicial a la que se desplace el vehículo que se quiera calcular, así, cuanto mayor sea esta velocidad, mayor será el paso del tiempo.

Los coeficientes de fricción también pueden variar en función de la velocidad, del mismo modo que lo hace la presión. El valor del coeficiente de fricción que se tomará para el cálculo será el inmediatamente superior al valor de la velocidad del vehículo. Existen hasta cinco coeficientes de fricción disponibles, *Friction Coeff 1*, *Friction Coeff 2*, *Friction Coeff 3*, *Friction Coeff 4* y *Friction Coeff 5*.

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

Initial Speed / Time step / Friction Coefficient				Vfin	Error	Speed (km/h)	Time Step	Friction Coeff 1	Friction Coeff 2	Friction Coeff 3	Friction Coeff 4	Friction Coeff 5
				0,00	0,05	120,00	0,15	0,35	0,31	0,00	0,35	0,35
						80,00	0,10	0,35	0,31	0,00	0,35	0,35
						70,00	0,10	0,35	0,31	0,00	0,35	0,35
						50,00	0,10	0,35	0,31	0,00	0,35	0,35
						40,00	0,05	0,35	0,31	0,00	0,35	0,35
						30,00	0,05	0,35	0,31	0,00	0,35	0,35
						20,00	0,05	0,35	0,31	0,00	0,35	0,35
						10,00	0,05	0,35	0,31	0,00	0,35	0,35
						0,00						

Figura 14. Velocidades iniciales, paso del tiempo, y coeficientes de fricción disponibles.

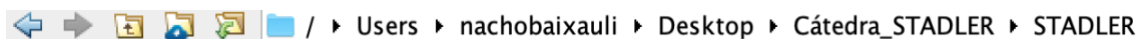
CAPÍTULO 5. MANUAL DE USUARIO

5.1 INSTALACIÓN DEL SOFTWARE

En primer lugar, el usuario ha de tener acceso al software MATLAB y disponer de una licencia para su uso. Para ello, se ha de instalar el software a través del siguiente enlace <https://es.mathworks.com/downloads/>.

Una vez instalada la versión recomendada en el apartado "requisitos del sistema", se ha de seleccionar la carpeta de trabajo adecuada que contenga las funciones, script y archivos que posteriormente se utilicen. En este proyecto se ha llamado a dicha carpeta de trabajo *STADLER*, y contiene los siguientes elementos, *DBF.m*, *degraded1.m*, *dynamic_mass.m*, *equal_len.m*, *export_cond1_txt.m*, *export_cond2_txt.m*, *export_cond3_txt.m*, *export_cond4_txt.m*, *export_cond5_txt.m*, *FBF.m*, *file2excel.m*, *forces_calculation.m*, *gen_vehicle_comp.m*, *initial_data.m*, *Input Data.xlsx*, *load_data.m*, *load_diam.m*, *main.m*, *nominal_mass.m*, *pressure_vel.m*, *result_cond1_load1_diam1.txt*, *result_cond2_load1_diam1.txt*, *result_cond3_load1_diam1.txt*, *result_cond4_load1_diam1.txt*, *result_cond5_load1_diam1.txt*, *Results.xlsx*, *TBK.m*, *vectors_zero.m*, *vectors.m*, *vehicle_resistance.m*, *wheel_diameter.m*.

Para seleccionar la carpeta de trabajo correspondiente se ha de presionar el icono de la carpeta con la flecha señalando hacia abajo y seleccionar la ruta de archivo *STADLER*.



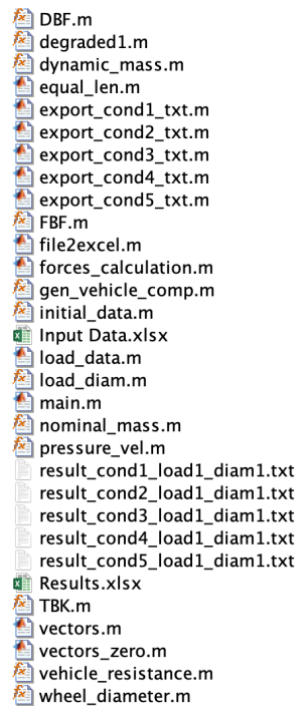


Figura 15. Elementos de la carpeta de trabajo *STADLER*.

5.2 EJECUCIÓN DEL PROGRAMA

Una vez instalado el software y seleccionada la carpeta de trabajo *STADLER*, se ha de editar el archivo de entrada *Input Data.xlsx* con las características del vehículo a calcular. Este apartado se hará manualmente por el usuario, abriendo el archivo *Input Data.xlsx*, editando los parámetros correspondientes, y guardando la versión al acabar. Como bien se ha comentado en el apartado "Fuente de datos", las dos primeras hojas del archivo contendrán las características del vehículo, mientras que la tercera hoja, 5.4, es la que el usuario podrá editar a su medida cada vez que vaya a ejecutar el programa, definiendo las condiciones degradadas a calcular.

Seguidamente, estando bien definidas las características del vehículo y las condiciones degradadas, se procede a ejecutar el programa en MATLAB. Para ello, se comienza ejecutando el script *load_data.m* que leerá las características del vehículo del archivo *Input Data.xlsx*.

Posteriormente, se ejecuta el script *main.m*. Antes de ejecutarse todo el código, el programa solicita al usuario que introduzca unos parámetros de entrada. Éstos son la velocidad inicial del vehículo, el paso del tiempo para cada iteración, y el número de condiciones degradadas que se

quiere calcular. Este es el paso que más tiempo consume, y dependiendo del número de condiciones degradadas que se quiera calcular, tardará entre 1 y 5 minutos.

```

Command Window
>> main
Which initial velocity you want to calculate?: 100
Which stp?: 0.15
fx How many conditions you want to calculate?: 5
    
```

Figura 16. Introducción de parámetros de entrada en MATLAB.

5.3 ANÁLISIS DE RESULTADOS

Finalizada la ejecución del script *main.m*, ya estarán todos los cálculos realizados y almacenados en sus respectivos archivos .txt. Para poder comprobar que los cálculos se han realizado correctamente, el usuario puede ejecutar el script *file2excel.m*. De esta manera, se exportará el archivo .txt, a elección del usuario a la plantilla de Excel *Results.xlsx*. Dicho archivo de Excel está diseñado para que por medio de gráficas se pueda visualizar la evolución de las fuerzas, velocidades, distancias, etc, con el tiempo, y poder detectar posibles anomalías.

El archivo de Excel *Results.xlsx* contiene dos hojas, *Data* y *Graphs*. La hoja *Data* contiene los resultados del cálculo en forma de vectores. Cada columna contiene un vector distinto, como bien se puede apreciar en la siguiente imagen.

velocity	time	pressure	Fc	Fb	nu	Fba	Fdyn	Fbrake	Fra	Fg	Fext	FB	Ftotal	acc	s
100	0	100	0	0	0,35	0	0	0	5505	0	0	0	5505	0,0802244	0
99,9567	0,15	100	0	0	0,35	0	0	0	5502,92	0	0	0	5502,92	0,0801941	4,1658
99,9134	0,3	100	0	0	0,35	0	3377,74	0	5500,84	0	0	3377,74	8878,58	0,129388	8,32979
99,8435	0,45	100	0	0	0,35	0	9717,8	0	5497,49	0	0	9717,8	15215,3	0,221733	12,4914
99,7238	0,6	100	0	0	0,35	0	17597,6	0	5491,75	0	0	17597,6	23089,4	0,336482	16,6491
99,5421	0,75	100	0	0	0,35	0	27038	0	5483,04	0	0	27038	32521	0,473929	20,8005
99,2861	0,9	100	0	0	0,35	0	38069,7	0	5470,77	0	0	38069,7	43540,5	0,634516	24,9428
98,9435	1,05	100	0	0	0,35	0	50736,4	0	5454,36	0	0	50736,4	56190,8	0,818868	29,0726
98,5013	1,2	100	0	0	0,35	0	55503,8	0	5433,19	0	0	55503,8	60937	0,888036	33,1861
98,0218	1,35	100	0	0	0,35	0	55775,4	0	5410,25	0	0	55775,4	61185,6	0,891659	37,2804
97,5403	1,5	100	0	0	0,35	0	56050,7	0	5387,23	0	0	56050,7	61437,9	0,895335	41,3546
97,0568	1,65	100	0	0	0,35	0	56329,9	0	5364,14	0	0	56329,9	61694	0,899068	45,4087
96,5713	1,8	100	0	0	0,35	0	56613,1	0	5340,98	0	0	56613,1	61954,1	0,902857	49,4427

Figura 17. Hoja Data del archivo *Results.xlsx*

Por otra parte, la hoja *Graphs* ofrece al usuario la posibilidad de visualizar gráficamente el contenido de los vectores del cálculo. Se han preestablecido unas gráficas comunes para la visualización de los resultados. Éstas graficas se muestran a continuación.

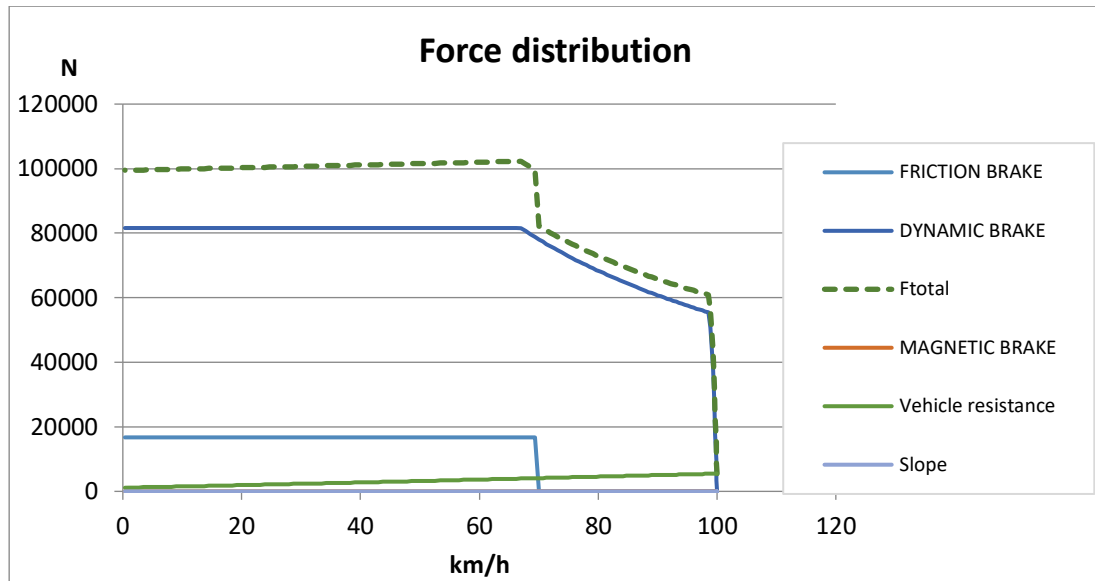


Figura 18. Valor de las fuerzas en función de la velocidad del vehículo

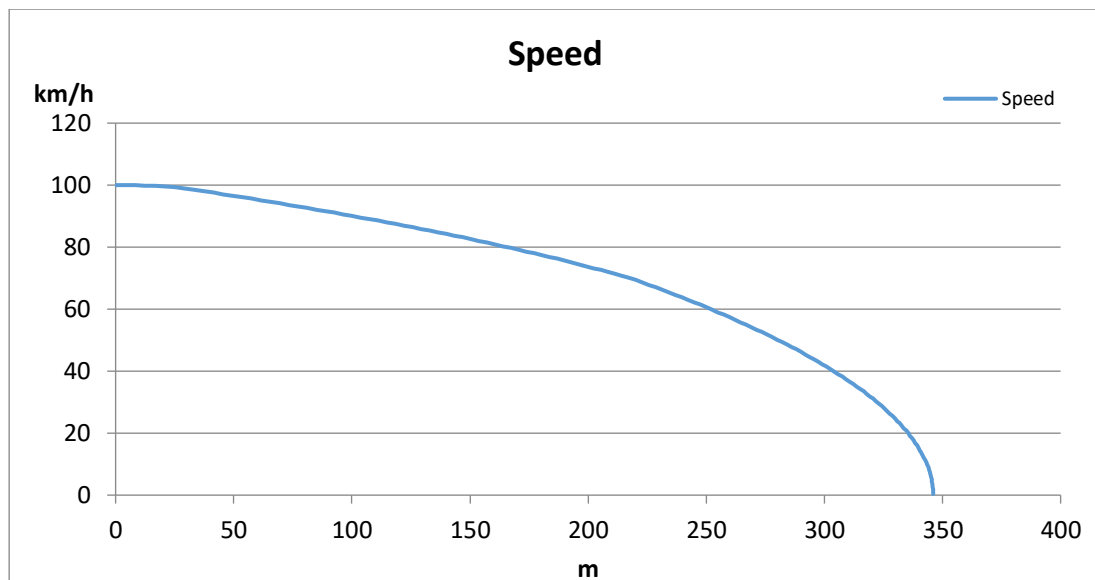


Figura 19. Velocidad del vehículo y distancia de frenado.

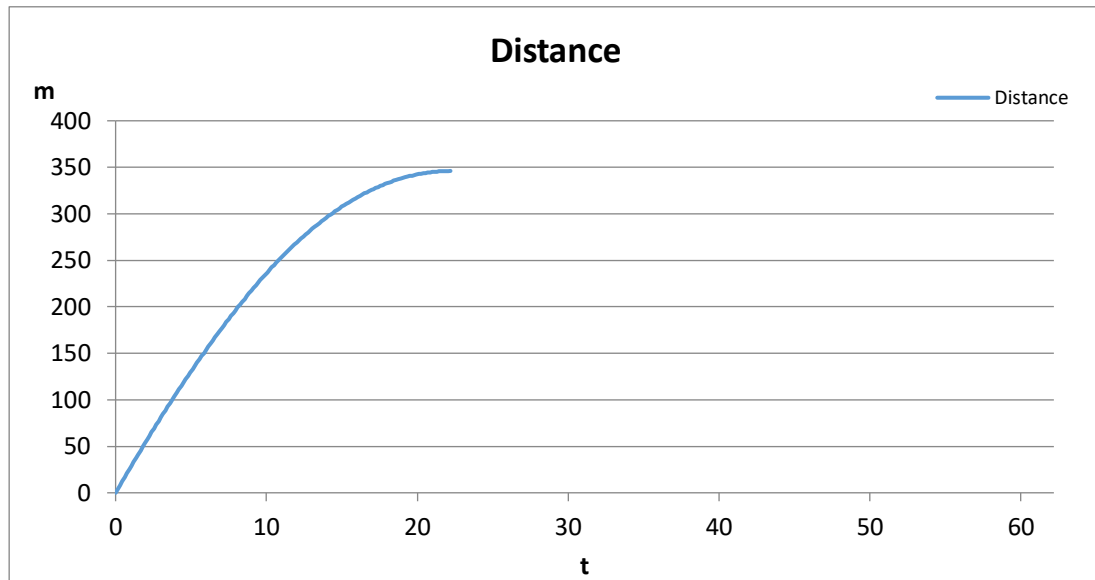


Figura 20. Distancia de frenado en función del tiempo.

Mediante estas gráficas el usuario del programa puede comprobar de una manera rápida y efectiva que sus cálculos no contienen ninguna anomalía, y, por consiguiente, pueden considerarse como válidos ya que en todo momento han seguido los patrones de cálculo establecidos por la norma UNE-EN 14531-2.

CAPÍTULO 6. PLAN DE DESARROLLO E IMPLEMENTACIÓN

El desarrollo e implementación de la herramienta ha seguido una estructura dividida en cuatro fases bien definidas, con una duración estimada de cuatro meses.

Fase/Mes	1	2	3	4
1. Revisión de la documentación y metodología.				
2. Programación de las fórmulas de la norma.				
3. Programación de las iteraciones de cálculo.				
4. Desarrollo de documentación y manuales.				

Tabla 5. Esquema de la duración del proyecto.

Inicialmente, el cliente solicitó que se diseñara una herramienta que evalúe las prestaciones del sistema de frenado con una eficiencia mayor a la de su herramienta anterior, y garantice que se cumplen los requisitos impuestos por la normativa UNE-EN 14531-2. De esta manera, reducir el tiempo de cálculo ha sido el foco de atención a lo largo de todo el proceso.

En la primera fase del proyecto se ha revisado en profundidad la documentación y metodología para llevar a cabo un cálculo tan complejo como lo es la evaluación de las prestaciones del sistema de frenado de un vehículo ferroviario. Al mismo tiempo, se ha optado por MATLAB como software para desarrollar la herramienta ya que ofrece un potente entorno de programación para desarrollo de algoritmos, análisis de datos, visualización y cálculo numérico, además de contener gráficas integradas que facilitan la visualización de los resultados y la obtención de información a partir de ellos. Una vez determinado el entorno de programación y las necesidades del cliente durante las primeras semanas del proyecto, se ha comenzado la segunda fase, la cual consiste en programar las fórmulas de la norma UNE-EN

14531-2.

Una vez comprendidas las necesidades del cliente, definido el entorno de desarrollo que se va a emplear, y estudiada la normativa a seguir, se ha procedido al desarrollo de las fórmulas necesarias para calcular el comportamiento de los frenos en unas situaciones dadas. Para ello se han creado distintos scripts y funciones que conjuntamente forman la herramienta.

Inicialmente se han creado varias funciones para almacenar en tablas la información recogida del archivo de entrada *Input Data*. Éstas funciones se ejecutan en el script *load_data*.

Posteriormente, se ha desarrollado el script *forces_calculation*, el cual contiene las fórmulas establecidas por la norma UNE-EN 14531-2. Se ha comenzado de esta manera para comprobar el rendimiento de cálculo del programa con tan solo una iteración, es decir, calculando tan solo una condición degradada.

Tanto la fase de revisión de la documentación y metodología como la fase de programación de las fórmulas de la norma han sido realizadas en el primer mes del proyecto.

Una vez diseñado el script *forces_calculation* con todos los cálculos a realizar para comprobar el rendimiento de los frenos del vehículo ferroviario, y comprobado que funciona, se ha proseguido a la creación de la tercera fase. Esta fase consiste en programar las iteraciones de cálculo, es decir, que el programa pueda calcular varias condiciones degradadas definidas por el usuario. Dicha fase ha sido la más laboriosa, pues requiere crear nuevos vectores y variables para cada condición degradada. Para ello, se han dedicado alrededor de dos meses para llevarlo a cabo.

Después de comprobar que el programa puede ejecutar correctamente las distintas condiciones degradadas definidas por el usuario y hacerlo en un tiempo considerablemente inferior al tiempo de cálculo de la hoja Excel, se ha creado la función *file2excel* para exportar los resultados a una plantilla Excel que nos permite visualizar gráficamente los resultados, como bien se ha comentado anteriormente. Para ello, inicialmente se ha creado el archivo *Results*, una plantilla de Excel que contiene unas gráficas preestablecidas capaces de mostrar si ha habido alguna anomalía en el cálculo.

Este último apartado, la creación tanto de la plantilla *Results* como de la función *file2excel*, se ha llevado a cabo en la última fase, la fase cuatro. En ella no sólo se ha acabado de desarrollar la herramienta, sino que también ha sido evaluada y testeada por los ingenieros de Stadler Rail para verificar su validez, al mismo tiempo que se ha llevado a cabo el desarrollo de la documentación de la misma.



CAPÍTULO 7. RESULTADO DEL PROYECTO

La aceptabilidad del resultado del cálculo realizado por la herramienta puede comprobarse mediante el coeficiente de desviación, un coeficiente obtenido de los cálculos de la distancia con pasos de frecuencia Δt y $(2 \cdot \Delta t)$. El valor del coeficiente de desviación de la distancia debe de ser como máximo de 0,02 con el fin de lograr un resultado aceptable para altas velocidades. El valor del coeficiente de la distancia viene expresado por la siguiente fórmula,

$$\Delta s = | s(2 \cdot \Delta t) \cdot s(\Delta t) | / | s(\Delta t) |$$

Δt – es el bucle de integración del paso del tiempo, [s];

$s(\Delta t)$ – es la distancia de frenado, calculada con el instante de tiempo Δt , en m;

$s(2 \cdot \Delta t)$ – es la distancia de frenado, calculada con el instante de tiempo $(2 \cdot \Delta t)$, en m;

Δs – es el coeficiente de desviación de la distancia, en m;

A modo de ensayo, se ha ejecutado el programa para calcular el sistema de freno y sus prestaciones para cinco condiciones degradadas, para así comprobar el tiempo de cálculo de la herramienta y su coeficiente de desviación. Estos valores, el tiempo de cálculo y el coeficiente de desviación, son los que van a marcar el éxito del proyecto. Una mejora del tiempo de cálculo respecto a la herramienta anterior constituye el fin último del presente trabajo de fin de grado.

El tiempo de cálculo de la herramienta anterior, para cinco condiciones degradadas distintas, se mantenía en dos horas aproximadas, pues una hoja Excel tiene un bajo rendimiento para realizar cálculos complejos. Sin embargo, la herramienta desarrollada en MATLAB consigue realizar el mismo cálculo en menos de un minuto. Tras realizar el ensayo se ha demostrado que es capaz de disminuir el tiempo de cálculo a 33 segundos, como bien se puede apreciar en la siguiente imagen.

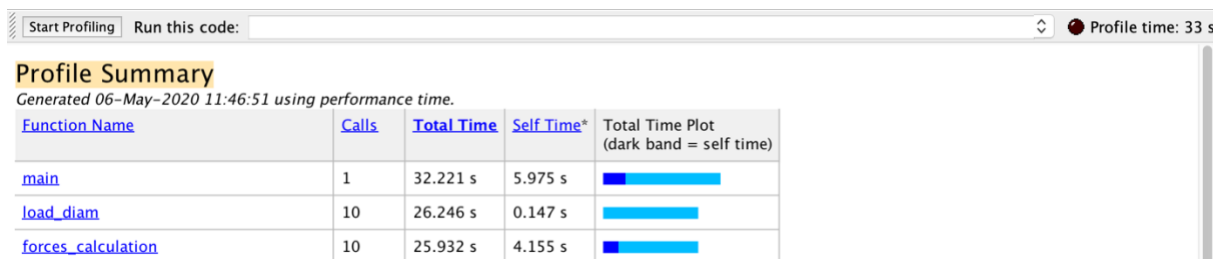


Figura 21. Tiempo de cálculo del programa.

De esta manera, se ha conseguido con el presente proyecto aumentar el rendimiento de cálculo un 21.800%, de un tiempo de cálculo de alrededor de 120 minutos a un tiempo de cálculo de 33 segundos.

El impacto económico que esto conlleva ha sido calculado a través del salario mensual de un titulado superior en la compañía Stadler Rail. Una vez conocido el salario mensual, se estima que un trabajador realiza una jornada laboral de 40 horas semanales, lo que multiplicado por cuatro semanas al mes suma 160 horas mensuales de trabajo. Así, si dividimos el salario mensual de un titulado superior entre 160 horas, tendremos el salario estimado del mismo por horas.

Como bien se puede apreciar en la siguiente tabla, el salario mensual de un titulado superior, nivel M1A, se mantiene en 1.975,09 euros. Esto son 12,34 euros por hora.

GRUPO	NIVEL	CATEGORÍA	SALARIOS
1	M1 A	TITULADO SUPERIOR	1.975,09
2	M2 A	TITULADO MEDIO	1.888,06
3	M3 A	PROYECTISTA JEFE	1.864,28
3	M3 B	JEFE TALLER	1.805,19
4	M4 A	MAESTRO TALLER	1.756,91

4	M4 B	CONTRAMAESTRE	1.710,04
4	M4 C	ENCARGADO	1.664,53
4	M4 D	A. T. T.	1.664,53
3	M3 C	JEFE 1ª ADMV.	1.633,17
4	M4 E	JEFE 2ª ADMV.	1.589,90

Tabla 6. Sueldos de convenio Stadler Rail.

Dado que se han reducido dos horas del trabajo de un titulado superior con el presente proyecto, se puede estimar que el ahorro económico gracias a esta herramienta se aproxima a 24,70 euros cada vez que se vaya a utilizar la misma. Se ha llegado a esta cifra multiplicando el coste/tiempo por el tiempo reducido, es decir, $12,35 \cdot 2 = 24,70$.

En definitiva, la compañía Stadler Rail Valencia ha valorado como exitoso este presente proyecto, pues se ha conseguido reducir considerablemente el tiempo dedicado al cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario y garantizar que se cumplen los requisitos impuestos por la normativa UNE-EN 14531-2, lo cual también repercute en un ahorro económico.

CAPÍTULO 8. CONCLUSIONES

Para concluir con el proyecto, se debe estudiar de qué forma se han logrado los objetivos expuestos en la introducción del proyecto.

En primer lugar, es destacable mencionar que el objetivo de desarrollar una herramienta que calcule las prestaciones del equipo de freno de un vehículo ferroviario reduciendo el tiempo de cálculo respecto a la herramienta anteriormente empleada y ajustándose a la normativa UNE-EN 14531-2 se ha logrado con creces. Esto se puede comprobar ya que tras el desarrollo de la herramienta se han hecho pruebas del funcionamiento de la misma y han sido satisfactorias. Además, se ha contado con la aprobación del departamento de frenos de la empresa Stadler Rail, por lo que queda demostrado que el proyecto ha sido exitoso.

Para conseguir el objetivo final, se han ido cumpliendo los objetivos específicos previamente establecidos al comienzo de este trabajo de fin de grado.

Se ha comenzado analizando la situación actual del cliente para definir las métricas que han determinado el éxito del proyecto, las cuales han sido el tiempo de cálculo y el coeficiente de desviación. A partir de las necesidades transmitidas por el cliente, se ha valorado el entorno de desarrollo que mejor se ajusta a estas mismas. Una vez definido MATLAB como entorno de trabajo, se ha desarrollado el código correspondiente para que el programa realice los cálculos necesarios para la evaluación del sistema de frenado del vehículo que se quiera diseñar.

Para el desarrollo de los cálculos, se ha seguido la normativa UNE-EN 14531-2, es decir, se han programado las ecuaciones de cálculo que establece dicha normativa. Se ha desarrollado, en primer lugar, el programa para el cálculo de una sola condición degradada, y una vez verificado su correcto funcionamiento, se ha añadido la posibilidad de realizar los cálculos para distintas condiciones degradadas.

Posteriormente al desarrollo de los cálculos establecidos por la normativa, se ha creado una plantilla que permite visualizar los resultados obtenidos para corroborar el correcto funcionamiento del programa.

Una vez terminada la herramienta, se ha testeado con el cálculo de una condición degradada para comparar el rendimiento de la herramienta creada con el rendimiento de la herramienta anterior. Como bien se ha comentado en el capítulo anterior, el resultado del proyecto ha sido valorado como exitoso al haberse reducido considerablemente el tiempo de cálculo con respecto

a la herramienta anterior, y haberse mantenido el coeficiente de desviación por debajo del máximo permitido.

CAPÍTULO 9. BIBLIOGRAFÍA

Signes Martínez, F. & Capilla Alpera, J. (2013). *El legado de la ingeniería valenciana. 115 años de los Devís a Vossloh (1897-2013)*. España, Valencia: Autoedición.

Romero, V. (2016). *De Macosa a Stadler: la fábrica de trenes que sobrevive a todas las crisis está en Albuixech*. Recuperado de El Confidencial sitio web:

https://www.elconfidencial.com/espana/comunidad-valenciana/2016-01-18/de-macosa-a-stadler-la-fabrica-de-trenes-que-sobrevive-a-todas-las-crisis-esta-en-albuixech_1136494/

Suárez, L. (2013). *Introducción a la programación en Matlab para ingenieros civiles y mecánicos*. United States: Createspace.

Báez López, D. & Cervantes Villagomez. O. (2016). *Matlab con aplicaciones a la ingeniería, física y finanzas*. Colombia: Alfaomega.

UNE-EN 14531-2. (2006). *Aplicaciones ferroviarias. Métodos para el cálculo de las distancias de frenado de parada y desaceleración e inmovilización*.

UNE-EN 14531-1. (2016). *Aplicaciones ferroviarias. Métodos para el cálculo de las distancias de parada, distancias de ralentización y frenado de inmovilización*.

STADLER RAIL VALENCIA S.A.U., VALENCIA, ESPAÑA. (2020). Recuperado de sitio web: <http://www.stadlerrail.com/es/sobre-nosotros>

Stadler Rail Valencia. *Documentos internos*. Valencia.

ANEXO I. PRESUPUESTO

El objetivo del presente documento anexo es exponer la valoración económica de este proyecto y servir como complemento a la memoria del mismo. Se han realizado todos los cálculos de costes del proyecto para su desarrollo e implementación. Para ello, se ha tenido en cuenta tanto el coste del personal de la empresa Stadler Rail que ha intervenido en el desarrollo de la herramienta, como el coste del estudiante involucrado en el mismo desarrollo y el coste del software empleado para ello.

I.I. CUADRO DE COSTES BÁSICOS

MANO DE OBRA			
Código	Unidad	Descripción	Precio Unitario
MA001	h	Graduado en Ingeniería en Tecnología Industriales	8,00 €
MA002	h	Ingeniero Superior	12,35 €

SOFTWARE			
Código	Unidad	Descripción	Precio Unitario
SW001	año	Licencia MATLAB	800,00 €

I.II. CUADRO DE PRECIOS DESCOMPUESTOS

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
UO 01 mes Desarrollo de herramienta de cálculo					
Tiempo dedicado al diseño y desarrollo de la herramienta de cálculo con el entorno de trabajo Matlab, realización de test y validación de los resultados.					
1		Mano de obra			
	h	Graduado en Ingeniería en Tecnología Industriales	100	8,00 €	800,00 €
	h	Ingeniero Superior	40	12,35 €	494,00 €
Subtotal mano de obra:					1.294,00 €
2		Software			
	año	Licencia MATLAB	1	800,00 €	800,00 €
Subtotal software:					800,00 €
Costes directos:					2.094,00 €

I.III. PRESUPUESTO TOTAL

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
UO 01	mes	Desarrollo de herramienta de cálculo	4	2.094,00 €	8.376,00 €
	%	Costes Indirectos	2	8.376,00 €	167,52 €
Presupuesto de Ejecución Material (PEM)					8.543,52 €
	%	Gastos Generales	10	8.543,52 €	854,35 €
Presupuesto de Ejecución por Contrata (PEC)					9.397,87 €
	%	Impuesto sobre el valor añadido (IVA)	21	9.397,87 €	1.973,55 €
Presupuesto Total					11.371,43 €

El presupuesto total del proyecto asciende a la cantidad de once mil trescientos setenta y un euros con cuarenta y tres céntimos.

ANEXO II. CÓDIGO

El objetivo del presente documento anexo es exponer el código desarrollado en este proyecto y servir como complemento a la memoria del mismo. Se han expuesto todas las funciones y scripts que constituyen la herramienta.

SCRIPT 1. LOAD_DATA

```
% function load_data()
% Reads data from the Excel file "Input Data.xlsx" and saves it in
% specific tables.

% Reads general characteristics data from excel.
%-----%
GeneralitiesVehicleComp_Data = gen_vehicle_comp("Input Data.xlsx",
"5.2", 4, 9);
NominalMass_Data = nominal_mass("Input Data.xlsx", "5.2", 12, 18);
DynamicMass_Data = dynamic_mass("Input Data.xlsx", "5.2", 20, 24);
WheelDiameter_Data = wheel_diameter("Input Data.xlsx", "5.2", 26, 29);
VehicleResistance_Data = vehicle_resistance("Input Data.xlsx", "5.2", 32,
36);
%-----%

% Reads brake type data from excel.
%-----%
FrictionBrakeForces_Data = FBF("Input Data.xlsx", "5.3", 5, 25);
DynamicBrakeForces_Data = DBF("Input Data.xlsx", "5.3", 30, 38);
TrackBrake_Data = TBK("Input Data.xlsx", "5.3", 43, 53);
RimBrakeForces_Data = FBF("Input Data.xlsx", "5.3", 58, 78);
%-----%

% Reads kinetic data from excel.
%-----%
Initial_Data = initial_data("Input Data.xlsx", "5.4", 4, 12);
Degraded_Data_1 = degraded1("Input Data.xlsx", "5.4", 5, 21);
Degraded_Data_2 = degraded1("Input Data.xlsx", "5.4", 25, 41);
Degraded_Data_3 = degraded1("Input Data.xlsx", "5.4", 45, 61);
Degraded_Data_4 = degraded1("Input Data.xlsx", "5.4", 65, 81);
Degraded_Data_5 = degraded1("Input Data.xlsx", "5.4", 85, 101);
Pressure_Vel_Data = pressure_vel("Input Data.xlsx", "5.4", 4, 23);
%-----%
```

SCRIPT 2. MAIN

```

% Ask for initial velocity, time step, and conditions to calculate
%-----%
prompt = 'Which initial velocity you want to calculate?: ';
x = input(prompt);

prompt = 'Which stp?: ';
stp = input(prompt);
stp2 = stp*2;

prompt = 'How many conditions you want to calculate?: ';
condit = input(prompt);

deviation = [];

for condition=1:condit

    if condition == 1
        sav = 0;

        [velocity1_load1_diame1,    time1_load1_diame1,
pressure1_load1_diame1,          Fc1_load1_diame1,    Fb1_load1_diame1,
nul_load1_diame1,                Fbal_load1_diame1,    Fdyn1_load1_diame1,
Fbrake1_load1_diame1,            Fra1_load1_diame1,    Fg1_load1_diame1,
Fext1_load1_diame1,             FB1_load1_diame1,     Ftotal1_load1_diame1,
acc1_load1_diame1,              s1_load1_diame1,      time_eq1_load1_diame1,
distance_eq1_load1_diame1,        deceleration_eq1_load1_diame1,
velocity1_load1_diame2,    time1_load1_diame2,    pressure1_load1_diame2,
Fc1_load1_diame2, Fb1_load1_diame2, nul_load1_diame2, Fbal_load1_diame2,
Fdyn1_load1_diame2, Fbrake1_load1_diame2, Fra1_load1_diame2,
Fg1_load1_diame2, Fext1_load1_diame2, FB1_load1_diame2,
Ftotal1_load1_diame2, acc1_load1_diame2, s1_load1_diame2,
time_eq1_load1_diame2,
distance_eq1_load1_diame2,        deceleration_eq1_load1_diame2,
velocity1_load1_diame3,    time1_load1_diame3,    pressure1_load1_diame3,
Fc1_load1_diame3, Fb1_load1_diame3, nul_load1_diame3, Fbal_load1_diame3,
Fdyn1_load1_diame3, Fbrake1_load1_diame3, Fra1_load1_diame3,
Fg1_load1_diame3, Fext1_load1_diame3, FB1_load1_diame3,
Ftotal1_load1_diame3, acc1_load1_diame3, s1_load1_diame3,
time_eq1_load1_diame3,
distance_eq1_load1_diame3,        deceleration_eq1_load1_diame3,
velocity1_load2_diame1,    time1_load2_diame1,    pressure1_load2_diame1,
Fc1_load2_diame1, Fb1_load2_diame1, nul_load2_diame1, Fbal_load2_diame1,
Fdyn1_load2_diame1, Fbrake1_load2_diame1, Fra1_load2_diame1,
Fg1_load2_diame1, Fext1_load2_diame1, FB1_load2_diame1,
Ftotal1_load2_diame1, acc1_load2_diame1, s1_load2_diame1,
time_eq1_load2_diame1,
distance_eq1_load2_diame1,        deceleration_eq1_load2_diame1,
velocity1_load2_diame2,    time1_load2_diame2,    pressure1_load2_diame2,
Fc1_load2_diame2, Fb1_load2_diame2, nul_load2_diame2, Fbal_load2_diame2,
Fdyn1_load2_diame2, Fbrake1_load2_diame2, Fra1_load2_diame2,
Fg1_load2_diame2, Fext1_load2_diame2, FB1_load2_diame2,

```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

Ftotal1_load2_diame2, acc1_load2_diame2, s1_load2_diame2,
time_eq1_load2_diame2,
distance_eq1_load2_diame2, deceleration_eq1_load2_diame2,
velocity1_load2_diame3, time1_load2_diame3, pressure1_load2_diame3,
Fc1_load2_diame3, Fb1_load2_diame3, nu1_load2_diame3, Fba1_load2_diame3,
Fdyn1_load2_diame3, Fbrake1_load2_diame3, Fra1_load2_diame3,
Fg1_load2_diame3, Fext1_load2_diame3, FB1_load2_diame3,
Ftotal1_load2_diame3, acc1_load2_diame3, s1_load2_diame3,
time_eq1_load2_diame3,
distance_eq1_load2_diame3, deceleration_eq1_load2_diame3,
velocity1_load3_diame1, time1_load3_diame1, pressure1_load3_diame1,
Fc1_load3_diame1, Fb1_load3_diame1, nu1_load3_diame1, Fba1_load3_diame1,
Fdyn1_load3_diame1, Fbrake1_load3_diame1, Fra1_load3_diame1,
Fg1_load3_diame1, Fext1_load3_diame1, FB1_load3_diame1,
Ftotal1_load3_diame1, acc1_load3_diame1, s1_load3_diame1,
time_eq1_load3_diame1,
distance_eq1_load3_diame1, deceleration_eq1_load3_diame1,
velocity1_load3_diame2, time1_load3_diame2, pressure1_load3_diame2,
Fc1_load3_diame2, Fb1_load3_diame2, nu1_load3_diame2, Fba1_load3_diame2,
Fdyn1_load3_diame2, Fbrake1_load3_diame2, Fra1_load3_diame2,
Fg1_load3_diame2, Fext1_load3_diame2, FB1_load3_diame2,
Ftotal1_load3_diame2, acc1_load3_diame2, s1_load3_diame2,
time_eq1_load3_diame2,
distance_eq1_load3_diame2, deceleration_eq1_load3_diame2,
velocity1_load3_diame3, time1_load3_diame3, pressure1_load3_diame3,
Fc1_load3_diame3, Fb1_load3_diame3, nu1_load3_diame3, Fba1_load3_diame3,
Fdyn1_load3_diame3, Fbrake1_load3_diame3, Fra1_load3_diame3,
Fg1_load3_diame3, Fext1_load3_diame3, FB1_load3_diame3,
Ftotal1_load3_diame3, acc1_load3_diame3, s1_load3_diame3,
time_eq1_load3_diame3,
distance_eq1_load3_diame3, deceleration_eq1_load3_diame3,
velocity1_load4_diame1, time1_load4_diame1, pressure1_load4_diame1,
Fc1_load4_diame1, Fb1_load4_diame1, nu1_load4_diame1, Fba1_load4_diame1,
Fdyn1_load4_diame1, Fbrake1_load4_diame1, Fra1_load4_diame1,
Fg1_load4_diame1, Fext1_load4_diame1, FB1_load4_diame1,
Ftotal1_load4_diame1, acc1_load4_diame1, s1_load4_diame1,
time_eq1_load4_diame1,
distance_eq1_load4_diame1, deceleration_eq1_load4_diame1,
velocity1_load4_diame2, time1_load4_diame2, pressure1_load4_diame2,
Fc1_load4_diame2, Fb1_load4_diame2, nu1_load4_diame2, Fba1_load4_diame2,
Fdyn1_load4_diame2, Fbrake1_load4_diame2, Fra1_load4_diame2,
Fg1_load4_diame2, Fext1_load4_diame2, FB1_load4_diame2,
Ftotal1_load4_diame2, acc1_load4_diame2, s1_load4_diame2,
time_eq1_load4_diame2,
distance_eq1_load4_diame2, deceleration_eq1_load4_diame2,
velocity1_load4_diame3, time1_load4_diame3, pressure1_load4_diame3,
Fc1_load4_diame3, Fb1_load4_diame3, nu1_load4_diame3, Fba1_load4_diame3,
Fdyn1_load4_diame3, Fbrake1_load4_diame3, Fra1_load4_diame3,
Fg1_load4_diame3, Fext1_load4_diame3, FB1_load4_diame3,
Ftotal1_load4_diame3, acc1_load4_diame3, s1_load4_diame3,
time_eq1_load4_diame3,
distance_eq1_load4_diame3, deceleration_eq1_load4_diame3, velocity1
_load5_diame1, time1_load5_diame1, pressure1_load5_diame1,
Fc1_load5_diame1, Fb1_load5_diame1, nu1_load5_diame1, Fba1_load5_diame1,
Fdyn1_load5_diame1, Fbrake1_load5_diame1, Fra1_load5_diame1,
Fg1_load5_diame1, Fext1_load5_diame1, FB1_load5_diame1,

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```

Ftotal1_load5_diame1,      acc1_load5_diame1,      s1_load5_diame1,
time_eq1_load5_diame1,
distance_eq1_load5_diame1,      deceleration_eq1_load5_diame1,
velocity1_load5_diame2,      time1_load5_diame2,      pressure1_load5_diame2,
Fc1_load5_diame2, Fb1_load5_diame2, nul_load5_diame2, Fba1_load5_diame2,
Fdyn1_load5_diame2,      Fbrake1_load5_diame2,      Fra1_load5_diame2,
Fg1_load5_diame2,      Fext1_load5_diame2,      FB1_load5_diame2,
Ftotal1_load5_diame2,      acc1_load5_diame2,      s1_load5_diame2,
time_eq1_load5_diame2,
distance_eq1_load5_diame2,      deceleration_eq1_load5_diame2,
velocity1_load5_diame3,      time1_load5_diame3,      pressure1_load5_diame3,
Fc1_load5_diame3, Fb1_load5_diame3, nul_load5_diame3, Fba1_load5_diame3,
Fdyn1_load5_diame3,      Fbrake1_load5_diame3,      Fra1_load5_diame3,
Fg1_load5_diame3,      Fext1_load5_diame3,      FB1_load5_diame3,
Ftotal1_load5_diame3,      acc1_load5_diame3,      s1_load5_diame3,
time_eq1_load5_diame3,
distance_eq1_load5_diame3,      deceleration_eq1_load5_diame3,      velocity1,
time1, pressure1, Fc1, Fb1, nul, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8,
Fdyn9, Fbrake1, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9,
Fg1, Fg2, Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext1, FB1, Ftotal1, acc1,
s1, time_eq1, distance_eq1, deceleration_eq1] = load_diam(stp2, x,
condition, sav, GeneralitiesVehicleComp_Data, NominalMass_Data,
DynamicMass_Data, WheelDiameter_Data, VehicleResistance_Data,
FrictionBrakeForces_Data, DynamicBrakeForces_Data, TrackBrake_Data,
Initial_Data, Degraded_Data_1, Pressure_Vel_Data);

```

```

s_load1_diame1_2 = s1_load1_diame1;
s_load1_diame2_2 = s1_load1_diame2;
s_load1_diame3_2 = s1_load1_diame3;
s_load2_diame1_2 = s1_load2_diame1;
s_load2_diame2_2 = s1_load2_diame2;
s_load2_diame3_2 = s1_load2_diame3;
s_load3_diame1_2 = s1_load3_diame1;
s_load3_diame2_2 = s1_load3_diame2;
s_load3_diame3_2 = s1_load3_diame3;
s_load4_diame1_2 = s1_load4_diame1;
s_load4_diame2_2 = s1_load4_diame2;
s_load4_diame3_2 = s1_load4_diame3;
s_load5_diame1_2 = s1_load5_diame1;
s_load5_diame2_2 = s1_load5_diame2;
s_load5_diame3_2 = s1_load5_diame3;

```

```

sav = 1;
[velocity1_load1_diame1,      time1_load1_diame1,
pressure1_load1_diame1,      Fc1_load1_diame1,      Fb1_load1_diame1,
nul_load1_diame1,      Fba1_load1_diame1,      Fdyn1_load1_diame1,
Fbrake1_load1_diame1,      Fra1_load1_diame1,      Fg1_load1_diame1,
Fext1_load1_diame1,      FB1_load1_diame1,      Ftotal1_load1_diame1,
acc1_load1_diame1,      s1_load1_diame1,      time_eq1_load1_diame1,
distance_eq1_load1_diame1,      deceleration_eq1_load1_diame1,
velocity1_load1_diame2,      time1_load1_diame2,      pressure1_load1_diame2,
Fc1_load1_diame2, Fb1_load1_diame2, nul_load1_diame2, Fba1_load1_diame2,
Fdyn1_load1_diame2,      Fbrake1_load1_diame2,      Fra1_load1_diame2,
Fg1_load1_diame2,      Fext1_load1_diame2,      FB1_load1_diame2,
Ftotal1_load1_diame2,      acc1_load1_diame2,      s1_load1_diame2,

```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
time_eq1_load1_diame2,
distance_eq1_load1_diame2,                deceleration_eq1_load1_diame2,
velocity1_load1_diame3,    time1_load1_diame3,    pressure1_load1_diame3,
Fc1_load1_diame3, Fb1_load1_diame3, nu1_load1_diame3, Fba1_load1_diame3,
Fdyn1_load1_diame3,    Fbrake1_load1_diame3,    Fra1_load1_diame3,
Fg1_load1_diame3,    Fext1_load1_diame3,    FB1_load1_diame3,
Ftotal1_load1_diame3,    acc1_load1_diame3,    s1_load1_diame3,
time_eq1_load1_diame3,
distance_eq1_load1_diame3,                deceleration_eq1_load1_diame3,
velocity1_load2_diame1,    time1_load2_diame1,    pressure1_load2_diame1,
Fc1_load2_diame1, Fb1_load2_diame1, nu1_load2_diame1, Fba1_load2_diame1,
Fdyn1_load2_diame1,    Fbrake1_load2_diame1,    Fra1_load2_diame1,
Fg1_load2_diame1,    Fext1_load2_diame1,    FB1_load2_diame1,
Ftotal1_load2_diame1,    acc1_load2_diame1,    s1_load2_diame1,
time_eq1_load2_diame1,
distance_eq1_load2_diame1,                deceleration_eq1_load2_diame1,
velocity1_load2_diame2,    time1_load2_diame2,    pressure1_load2_diame2,
Fc1_load2_diame2, Fb1_load2_diame2, nu1_load2_diame2, Fba1_load2_diame2,
Fdyn1_load2_diame2,    Fbrake1_load2_diame2,    Fra1_load2_diame2,
Fg1_load2_diame2,    Fext1_load2_diame2,    FB1_load2_diame2,
Ftotal1_load2_diame2,    acc1_load2_diame2,    s1_load2_diame2,
time_eq1_load2_diame2,
distance_eq1_load2_diame2,                deceleration_eq1_load2_diame2,
velocity1_load2_diame3,    time1_load2_diame3,    pressure1_load2_diame3,
Fc1_load2_diame3, Fb1_load2_diame3, nu1_load2_diame3, Fba1_load2_diame3,
Fdyn1_load2_diame3,    Fbrake1_load2_diame3,    Fra1_load2_diame3,
Fg1_load2_diame3,    Fext1_load2_diame3,    FB1_load2_diame3,
Ftotal1_load2_diame3,    acc1_load2_diame3,    s1_load2_diame3,
time_eq1_load2_diame3,
distance_eq1_load2_diame3,                deceleration_eq1_load2_diame3,
velocity1_load3_diame1,    time1_load3_diame1,    pressure1_load3_diame1,
Fc1_load3_diame1, Fb1_load3_diame1, nu1_load3_diame1, Fba1_load3_diame1,
Fdyn1_load3_diame1,    Fbrake1_load3_diame1,    Fra1_load3_diame1,
Fg1_load3_diame1,    Fext1_load3_diame1,    FB1_load3_diame1,
Ftotal1_load3_diame1,    acc1_load3_diame1,    s1_load3_diame1,
time_eq1_load3_diame1,
distance_eq1_load3_diame1,                deceleration_eq1_load3_diame1,
velocity1_load3_diame2,    time1_load3_diame2,    pressure1_load3_diame2,
Fc1_load3_diame2, Fb1_load3_diame2, nu1_load3_diame2, Fba1_load3_diame2,
Fdyn1_load3_diame2,    Fbrake1_load3_diame2,    Fra1_load3_diame2,
Fg1_load3_diame2,    Fext1_load3_diame2,    FB1_load3_diame2,
Ftotal1_load3_diame2,    acc1_load3_diame2,    s1_load3_diame2,
time_eq1_load3_diame2,
distance_eq1_load3_diame2,                deceleration_eq1_load3_diame2,
velocity1_load3_diame3,    time1_load3_diame3,    pressure1_load3_diame3,
Fc1_load3_diame3, Fb1_load3_diame3, nu1_load3_diame3, Fba1_load3_diame3,
Fdyn1_load3_diame3,    Fbrake1_load3_diame3,    Fra1_load3_diame3,
Fg1_load3_diame3,    Fext1_load3_diame3,    FB1_load3_diame3,
Ftotal1_load3_diame3,    acc1_load3_diame3,    s1_load3_diame3,
time_eq1_load3_diame3,
distance_eq1_load3_diame3,                deceleration_eq1_load3_diame3,
velocity1_load4_diame1,    time1_load4_diame1,    pressure1_load4_diame1,
Fc1_load4_diame1, Fb1_load4_diame1, nu1_load4_diame1, Fba1_load4_diame1,
Fdyn1_load4_diame1,    Fbrake1_load4_diame1,    Fra1_load4_diame1,
Fg1_load4_diame1,    Fext1_load4_diame1,    FB1_load4_diame1,
Ftotal1_load4_diame1,    acc1_load4_diame1,    s1_load4_diame1,
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
time_eq1_load4_diame1,
distance_eq1_load4_diame1,          deceleration_eq1_load4_diame1,
velocity1_load4_diame2,    time1_load4_diame2,    pressure1_load4_diame2,
Fc1_load4_diame2, Fb1_load4_diame2, nul_load4_diame2, Fba1_load4_diame2,
Fdyn1_load4_diame2,    Fbrake1_load4_diame2,    Fra1_load4_diame2,
Fg1_load4_diame2,    Fext1_load4_diame2,    FB1_load4_diame2,
Ftotal1_load4_diame2,    acc1_load4_diame2,    s1_load4_diame2,
time_eq1_load4_diame2,
distance_eq1_load4_diame2,          deceleration_eq1_load4_diame2,
velocity1_load4_diame3,    time1_load4_diame3,    pressure1_load4_diame3,
Fc1_load4_diame3, Fb1_load4_diame3, nul_load4_diame3, Fba1_load4_diame3,
Fdyn1_load4_diame3,    Fbrake1_load4_diame3,    Fra1_load4_diame3,
Fg1_load4_diame3,    Fext1_load4_diame3,    FB1_load4_diame3,
Ftotal1_load4_diame3,    acc1_load4_diame3,    s1_load4_diame3,
time_eq1_load4_diame3,
distance_eq1_load4_diame3,    deceleration_eq1_load4_diame3,    velocity1
_load5_diame1,    time1_load5_diame1,    pressure1_load5_diame1,
Fc1_load5_diame1, Fb1_load5_diame1, nul_load5_diame1, Fba1_load5_diame1,
Fdyn1_load5_diame1,    Fbrake1_load5_diame1,    Fra1_load5_diame1,
Fg1_load5_diame1,    Fext1_load5_diame1,    FB1_load5_diame1,
Ftotal1_load5_diame1,    acc1_load5_diame1,    s1_load5_diame1,
time_eq1_load5_diame1,
distance_eq1_load5_diame1,          deceleration_eq1_load5_diame1,
velocity1_load5_diame2,    time1_load5_diame2,    pressure1_load5_diame2,
Fc1_load5_diame2, Fb1_load5_diame2, nul_load5_diame2, Fba1_load5_diame2,
Fdyn1_load5_diame2,    Fbrake1_load5_diame2,    Fra1_load5_diame2,
Fg1_load5_diame2,    Fext1_load5_diame2,    FB1_load5_diame2,
Ftotal1_load5_diame2,    acc1_load5_diame2,    s1_load5_diame2,
time_eq1_load5_diame2,
distance_eq1_load5_diame2,          deceleration_eq1_load5_diame2,
velocity1_load5_diame3,    time1_load5_diame3,    pressure1_load5_diame3,
Fc1_load5_diame3, Fb1_load5_diame3, nul_load5_diame3, Fba1_load5_diame3,
Fdyn1_load5_diame3,    Fbrake1_load5_diame3,    Fra1_load5_diame3,
Fg1_load5_diame3,    Fext1_load5_diame3,    FB1_load5_diame3,
Ftotal1_load5_diame3,    acc1_load5_diame3,    s1_load5_diame3,
time_eq1_load5_diame3,
distance_eq1_load5_diame3,    deceleration_eq1_load5_diame3,    velocity1,
time1, pressure1, Fc1, Fb1, nul, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8,
Fdyn9, Fbrake1, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9,
Fg1, Fg2, Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext1, FB1, Ftotal1, acc1,
s1, time_eq1, distance_eq1, deceleration_eq1] = load_diam(stp, x,
condition, sav, GeneralitiesVehicleComp_Data, NominalMass_Data,
DynamicMass_Data, WheelDiameter_Data, VehicleResistance_Data,
FrictionBrakeForces_Data, DynamicBrakeForces_Data, TrackBrake_Data,
Initial_Data, Degraded_Data_1, Pressure_Vel_Data);

Fba1_cond1 = Fba1;
Fba2_cond1 = Fba2;
Fba3_cond1 = Fba3;
Fba4_cond1 = Fba4;
Fba5_cond1 = Fba5;
Fba6_cond1 = Fba6;
Fba7_cond1 = Fba7;
Fba8_cond1 = Fba8;
Fba9_cond1 = Fba9;
```

```
Fdyn1_cond1 = Fdyn1;
Fdyn2_cond1 = Fdyn2;
Fdyn3_cond1 = Fdyn3;
Fdyn4_cond1 = Fdyn4;
Fdyn5_cond1 = Fdyn5;
Fdyn6_cond1 = Fdyn6;
Fdyn7_cond1 = Fdyn7;
Fdyn8_cond1 = Fdyn8;
Fdyn9_cond1 = Fdyn9;

Fra1_cond1 = Fra1;
Fra2_cond1 = Fra2;
Fra3_cond1 = Fra3;
Fra4_cond1 = Fra4;
Fra5_cond1 = Fra5;
Fra6_cond1 = Fra6;
Fra7_cond1 = Fra7;
Fra8_cond1 = Fra8;
Fra9_cond1 = Fra9;

Fg1_cond1 = Fg1;
Fg2_cond1 = Fg2;
Fg3_cond1 = Fg3;
Fg4_cond1 = Fg4;
Fg5_cond1 = Fg5;
Fg6_cond1 = Fg6;
Fg7_cond1 = Fg7;
Fg8_cond1 = Fg8;
Fg9_cond1 = Fg9;

s_load1_diame1_1 = s1_load1_diame1;
s_load1_diame2_1 = s1_load1_diame2;
s_load1_diame3_1 = s1_load1_diame3;
s_load2_diame1_1 = s1_load2_diame1;
s_load2_diame2_1 = s1_load2_diame2;
s_load2_diame3_1 = s1_load2_diame3;
s_load3_diame1_1 = s1_load3_diame1;
s_load3_diame2_1 = s1_load3_diame2;
s_load3_diame3_1 = s1_load3_diame3;
s_load4_diame1_1 = s1_load4_diame1;
s_load4_diame2_1 = s1_load4_diame2;
s_load4_diame3_1 = s1_load4_diame3;
s_load5_diame1_1 = s1_load5_diame1;
s_load5_diame2_1 = s1_load5_diame2;
s_load5_diame3_1 = s1_load5_diame3;

if length(s_load1_diame1_2) ~= 0
    dev = abs((s_load1_diame1_2(end) -
s_load1_diame1_1(end))/s_load1_diame1_1(end));
    deviation1_load1_diame1(condition) = dev;
end
if length(s_load1_diame2_2) ~= 0
    dev = abs((s_load1_diame2_2(end) -
s_load1_diame2_1(end))/s_load1_diame2_1(end));
    deviation1_load1_diame2(condition) = dev;
```

```
end
if length(s_load1_diame3_2) ~= 0
    dev = abs((s_load1_diame3_2(end) -
s_load1_diame3_1(end))/s_load1_diame3_1(end));
    deviation1_load1_diame3(condition) = dev;
end
if length(s_load2_diame1_2) ~= 0
    dev = abs((s_load2_diame1_2(end) -
s_load2_diame1_1(end))/s_load2_diame1_1(end));
    deviation1_load2_diame1(condition) = dev;
end
if length(s_load2_diame2_2) ~= 0
    dev = abs((s_load2_diame2_2(end) -
s_load2_diame2_1(end))/s_load2_diame2_1(end));
    deviation1_load2_diame2(condition) = dev;
end
if length(s_load2_diame3_2) ~= 0
    dev = abs((s_load2_diame3_2(end) -
s_load2_diame3_1(end))/s_load2_diame3_1(end));
    deviation1_load2_diame3(condition) = dev;
end
if length(s_load3_diame1_2) ~= 0
    dev = abs((s_load3_diame1_2(end) -
s_load3_diame1_1(end))/s_load3_diame1_1(end));
    deviation1_load3_diame1(condition) = dev;
end
if length(s_load3_diame2_2) ~= 0
    dev = abs((s_load3_diame2_2(end) -
s_load3_diame2_1(end))/s_load3_diame2_1(end));
    deviation1_load3_diame2(condition) = dev;
end
if length(s_load3_diame3_2) ~= 0
    dev = abs((s_load3_diame3_2(end) -
s_load3_diame3_1(end))/s_load3_diame3_1(end));
    deviation1_load3_diame3(condition) = dev;
end
if length(s_load4_diame1_2) ~= 0
    dev = abs((s_load4_diame1_2(end) -
s_load4_diame1_1(end))/s_load4_diame1_1(end));
    deviation1_load4_diame1(condition) = dev;
end
if length(s_load4_diame2_2) ~= 0
    dev = abs((s_load4_diame2_2(end) -
s_load4_diame2_1(end))/s_load4_diame2_1(end));
    deviation1_load4_diame2(condition) = dev;
end
if length(s_load4_diame3_2) ~= 0
    dev = abs((s_load4_diame3_2(end) -
s_load4_diame3_1(end))/s_load4_diame3_1(end));
    deviation1_load4_diame3(condition) = dev;
end
if length(s_load5_diame1_2) ~= 0
    dev = abs((s_load5_diame1_2(end) -
s_load5_diame1_1(end))/s_load5_diame1_1(end));
    deviation1_load5_diame1(condition) = dev;
end
```


**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```

        if length(s_load5_diame2_2) ~= 0
            dev = abs((s_load5_diame2_2(end) -
s_load5_diame2_1(end))/s_load5_diame2_1(end));
            deviation1_load5_diame2(condition) = dev;
        end
        if length(s_load5_diame3_2) ~= 0
            dev = abs((s_load5_diame3_2(end) -
s_load5_diame3_1(end))/s_load5_diame3_1(end));
            deviation1_load5_diame3(condition) = dev;
        end
    end
    if condition == 2
        sav = 0;

        [velocity2_load1_diame1,    time2_load1_diame1,
pressure2_load1_diame1,    Fc2_load1_diame1,    Fb2_load1_diame1,
nu2_load1_diame1,    Fba2_load1_diame1,    Fdyn2_load1_diame1,
Fbrake2_load1_diame1,    Fra2_load1_diame1,    Fg2_load1_diame1,
Fext2_load1_diame1,    Fb2_load1_diame1,    Ftotal2_load1_diame1,
acc2_load1_diame1,    s2_load1_diame1,    time_eq2_load1_diame1,
distance_eq2_load1_diame1,    deceleration_eq2_load1_diame1,
velocity2_load1_diame2,    time2_load1_diame2,    pressure2_load1_diame2,
Fc2_load1_diame2,    Fb2_load1_diame2,    nu2_load1_diame2,    Fba2_load1_diame2,
Fdyn2_load1_diame2,    Fbrake2_load1_diame2,    Fra2_load1_diame2,
Fg2_load1_diame2,    Fext2_load1_diame2,    Fb2_load1_diame2,
Ftotal2_load1_diame2,    acc2_load1_diame2,    s2_load1_diame2,
time_eq2_load1_diame2,
distance_eq2_load1_diame2,    deceleration_eq2_load1_diame2,
velocity2_load1_diame3,    time2_load1_diame3,    pressure2_load1_diame3,
Fc2_load1_diame3,    Fb2_load1_diame3,    nu2_load1_diame3,    Fba2_load1_diame3,
Fdyn2_load1_diame3,    Fbrake2_load1_diame3,    Fra2_load1_diame3,
Fg2_load1_diame3,    Fext2_load1_diame3,    Fb2_load1_diame3,
Ftotal2_load1_diame3,    acc2_load1_diame3,    s2_load1_diame3,
time_eq2_load1_diame3,
distance_eq2_load1_diame3,    deceleration_eq2_load1_diame3,
velocity2_load2_diame1,    time2_load2_diame1,    pressure2_load2_diame1,
Fc2_load2_diame1,    Fb2_load2_diame1,    nu2_load2_diame1,    Fba2_load2_diame1,
Fdyn2_load2_diame1,    Fbrake2_load2_diame1,    Fra2_load2_diame1,
Fg2_load2_diame1,    Fext2_load2_diame1,    Fb2_load2_diame1,
Ftotal2_load2_diame1,    acc2_load2_diame1,    s2_load2_diame1,
time_eq2_load2_diame1,
distance_eq2_load2_diame1,    deceleration_eq2_load2_diame1,
velocity2_load2_diame2,    time2_load2_diame2,    pressure2_load2_diame2,
Fc2_load2_diame2,    Fb2_load2_diame2,    nu2_load2_diame2,    Fba2_load2_diame2,
Fdyn2_load2_diame2,    Fbrake2_load2_diame2,    Fra2_load2_diame2,
Fg2_load2_diame2,    Fext2_load2_diame2,    Fb2_load2_diame2,
Ftotal2_load2_diame2,    acc2_load2_diame2,    s2_load2_diame2,
time_eq2_load2_diame2,
distance_eq2_load2_diame2,    deceleration_eq2_load2_diame2,
velocity2_load2_diame3,    time2_load2_diame3,    pressure2_load2_diame3,
Fc2_load2_diame3,    Fb2_load2_diame3,    nu2_load2_diame3,    Fba2_load2_diame3,
Fdyn2_load2_diame3,    Fbrake2_load2_diame3,    Fra2_load2_diame3,
Fg2_load2_diame3,    Fext2_load2_diame3,    Fb2_load2_diame3,
Ftotal2_load2_diame3,    acc2_load2_diame3,    s2_load2_diame3,
time_eq2_load2_diame3,
distance_eq2_load2_diame3,    deceleration_eq2_load2_diame3,
velocity2_load3_diame1,    time2_load3_diame1,    pressure2_load3_diame1,

```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

Fc2_load3_diame1, Fb2_load3_diame1, nu2_load3_diame1, Fba2_load3_diame1,
Fdyn2_load3_diame1, Fbrake2_load3_diame1, Fra2_load3_diame1,
Fg2_load3_diame1, Fext2_load3_diame1, Fb2_load3_diame1,
Ftotal2_load3_diame1, acc2_load3_diame1, s2_load3_diame1,
time_eq2_load3_diame1,
distance_eq2_load3_diame1, deceleration_eq2_load3_diame1,
velocity2_load3_diame2, time2_load3_diame2, pressure2_load3_diame2,
Fc2_load3_diame2, Fb2_load3_diame2, nu2_load3_diame2, Fba2_load3_diame2,
Fdyn2_load3_diame2, Fbrake2_load3_diame2, Fra2_load3_diame2,
Fg2_load3_diame2, Fext2_load3_diame2, Fb2_load3_diame2,
Ftotal2_load3_diame2, acc2_load3_diame2, s2_load3_diame2,
time_eq2_load3_diame2,
distance_eq2_load3_diame2, deceleration_eq2_load3_diame2,
velocity2_load3_diame3, time2_load3_diame3, pressure2_load3_diame3,
Fc2_load3_diame3, Fb2_load3_diame3, nu2_load3_diame3, Fba2_load3_diame3,
Fdyn2_load3_diame3, Fbrake2_load3_diame3, Fra2_load3_diame3,
Fg2_load3_diame3, Fext2_load3_diame3, Fb2_load3_diame3,
Ftotal2_load3_diame3, acc2_load3_diame3, s2_load3_diame3,
time_eq2_load3_diame3,
distance_eq2_load3_diame3, deceleration_eq2_load3_diame3,
velocity2_load4_diame1, time2_load4_diame1, pressure2_load4_diame1,
Fc2_load4_diame1, Fb2_load4_diame1, nu2_load4_diame1, Fba2_load4_diame1,
Fdyn2_load4_diame1, Fbrake2_load4_diame1, Fra2_load4_diame1,
Fg2_load4_diame1, Fext2_load4_diame1, Fb2_load4_diame1,
Ftotal2_load4_diame1, acc2_load4_diame1, s2_load4_diame1,
time_eq2_load4_diame1,
distance_eq2_load4_diame1, deceleration_eq2_load4_diame1,
velocity2_load4_diame2, time2_load4_diame2, pressure2_load4_diame2,
Fc2_load4_diame2, Fb2_load4_diame2, nu2_load4_diame2, Fba2_load4_diame2,
Fdyn2_load4_diame2, Fbrake2_load4_diame2, Fra2_load4_diame2,
Fg2_load4_diame2, Fext2_load4_diame2, Fb2_load4_diame2,
Ftotal2_load4_diame2, acc2_load4_diame2, s2_load4_diame2,
time_eq2_load4_diame2,
distance_eq2_load4_diame2, deceleration_eq2_load4_diame2,
velocity2_load4_diame3, time2_load4_diame3, pressure2_load4_diame3,
Fc2_load4_diame3, Fb2_load4_diame3, nu2_load4_diame3, Fba2_load4_diame3,
Fdyn2_load4_diame3, Fbrake2_load4_diame3, Fra2_load4_diame3,
Fg2_load4_diame3, Fext2_load4_diame3, Fb2_load4_diame3,
Ftotal2_load4_diame3, acc2_load4_diame3, s2_load4_diame3,
time_eq2_load4_diame3,
distance_eq2_load4_diame3, deceleration_eq2_load4_diame3, velocity2
_load5_diame1, time2_load5_diame1, pressure2_load5_diame1,
Fc2_load5_diame1, Fb2_load5_diame1, nu2_load5_diame1, Fba2_load5_diame1,
Fdyn2_load5_diame1, Fbrake2_load5_diame1, Fra2_load5_diame1,
Fg2_load5_diame1, Fext2_load5_diame1, Fb2_load5_diame1,
Ftotal2_load5_diame1, acc2_load5_diame1, s2_load5_diame1,
time_eq2_load5_diame1,
distance_eq2_load5_diame1, deceleration_eq2_load5_diame1,
velocity2_load5_diame2, time2_load5_diame2, pressure2_load5_diame2,
Fc2_load5_diame2, Fb2_load5_diame2, nu2_load5_diame2, Fba2_load5_diame2,
Fdyn2_load5_diame2, Fbrake2_load5_diame2, Fra2_load5_diame2,
Fg2_load5_diame2, Fext2_load5_diame2, Fb2_load5_diame2,
Ftotal2_load5_diame2, acc2_load5_diame2, s2_load5_diame2,
time_eq2_load5_diame2,
distance_eq2_load5_diame2, deceleration_eq2_load5_diame2,
velocity2_load5_diame3, time2_load5_diame3, pressure2_load5_diame3,

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```
Fc2_load5_diame3, Fb2_load5_diame3, nu2_load5_diame3, Fba2_load5_diame3,  
Fdyn2_load5_diame3, Fbrake2_load5_diame3, Fra2_load5_diame3,  
Fg2_load5_diame3, Fext2_load5_diame3, Fb2_load5_diame3,  
Ftotal2_load5_diame3, acc2_load5_diame3, s2_load5_diame3,  
time_eq2_load5_diame3,  
distance_eq2_load5_diame3, deceleration_eq2_load5_diame3, velocity2,  
time2, pressure2, Fc2, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7, Fba8,  
Fba9, nu2, Fba2, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8,  
Fdyn9, Fbrake2, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9,  
Fg1, Fg2, Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext2, FB2, Ftotal2, acc2,  
s2, time_eq2, distance_eq2, deceleration_eq2] = load_diam(stp2, x,  
condition, sav, GeneralitiesVehicleComp_Data, NominalMass_Data,  
DynamicMass_Data, WheelDiameter_Data, VehicleResistance_Data,  
FrictionBrakeForces_Data, DynamicBrakeForces_Data, TrackBrake_Data,  
Initial_Data, Degraded_Data_2, Pressure_Vel_Data);
```

```
s_load1_diame1_2 = s2_load1_diame1;  
s_load1_diame2_2 = s2_load1_diame2;  
s_load1_diame3_2 = s2_load1_diame3;  
s_load2_diame1_2 = s2_load2_diame1;  
s_load2_diame2_2 = s2_load2_diame2;  
s_load2_diame3_2 = s2_load2_diame3;  
s_load3_diame1_2 = s2_load3_diame1;  
s_load3_diame2_2 = s2_load3_diame2;  
s_load3_diame3_2 = s2_load3_diame3;  
s_load4_diame1_2 = s2_load4_diame1;  
s_load4_diame2_2 = s2_load4_diame2;  
s_load4_diame3_2 = s2_load4_diame3;  
s_load5_diame1_2 = s2_load5_diame1;  
s_load5_diame2_2 = s2_load5_diame2;  
s_load5_diame3_2 = s2_load5_diame3;
```

```
sav = 1;  
[velocity2_load1_diame1, time2_load1_diame1,  
pressure2_load1_diame1, Fc2_load1_diame1, Fb2_load1_diame1,  
nu2_load1_diame1, Fba2_load1_diame1, Fdyn2_load1_diame1,  
Fbrake2_load1_diame1, Fra2_load1_diame1, Fg2_load1_diame1,  
Fext2_load1_diame1, Fb2_load1_diame1, Ftotal2_load1_diame1,  
acc2_load1_diame1, s2_load1_diame1, time_eq2_load1_diame1,  
distance_eq2_load1_diame1, deceleration_eq2_load1_diame1,  
velocity2_load1_diame2, time2_load1_diame2, pressure2_load1_diame2,  
Fc2_load1_diame2, Fb2_load1_diame2, nu2_load1_diame2, Fba2_load1_diame2,  
Fdyn2_load1_diame2, Fbrake2_load1_diame2, Fra2_load1_diame2,  
Fg2_load1_diame2, Fext2_load1_diame2, Fb2_load1_diame2,  
Ftotal2_load1_diame2, acc2_load1_diame2, s2_load1_diame2,  
time_eq2_load1_diame2, deceleration_eq2_load1_diame2,  
velocity2_load1_diame3, time2_load1_diame3, pressure2_load1_diame3,  
Fc2_load1_diame3, Fb2_load1_diame3, nu2_load1_diame3, Fba2_load1_diame3,  
Fdyn2_load1_diame3, Fbrake2_load1_diame3, Fra2_load1_diame3,  
Fg2_load1_diame3, Fext2_load1_diame3, Fb2_load1_diame3,  
Ftotal2_load1_diame3, acc2_load1_diame3, s2_load1_diame3,  
time_eq2_load1_diame3, deceleration_eq2_load1_diame3,  
velocity2_load2_diame1, time2_load2_diame1, pressure2_load2_diame1,  
Fc2_load2_diame1, Fb2_load2_diame1, nu2_load2_diame1, Fba2_load2_diame1,
```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

Fdyn2_load2_diame1, Fbrake2_load2_diame1, Fra2_load2_diame1,
Fg2_load2_diame1, Fext2_load2_diame1, Fb2_load2_diame1,
Ftotal2_load2_diame1, acc2_load2_diame1, s2_load2_diame1,
time_eq2_load2_diame1,
distance_eq2_load2_diame1, deceleration_eq2_load2_diame1,
velocity2_load2_diame2, time2_load2_diame2, pressure2_load2_diame2,
Fc2_load2_diame2, Fb2_load2_diame2, nu2_load2_diame2, Fba2_load2_diame2,
Fdyn2_load2_diame2, Fbrake2_load2_diame2, Fra2_load2_diame2,
Fg2_load2_diame2, Fext2_load2_diame2, Fb2_load2_diame2,
Ftotal2_load2_diame2, acc2_load2_diame2, s2_load2_diame2,
time_eq2_load2_diame2,
distance_eq2_load2_diame2, deceleration_eq2_load2_diame2,
velocity2_load2_diame3, time2_load2_diame3, pressure2_load2_diame3,
Fc2_load2_diame3, Fb2_load2_diame3, nu2_load2_diame3, Fba2_load2_diame3,
Fdyn2_load2_diame3, Fbrake2_load2_diame3, Fra2_load2_diame3,
Fg2_load2_diame3, Fext2_load2_diame3, Fb2_load2_diame3,
Ftotal2_load2_diame3, acc2_load2_diame3, s2_load2_diame3,
time_eq2_load2_diame3,
distance_eq2_load2_diame3, deceleration_eq2_load2_diame3,
velocity2_load3_diame1, time2_load3_diame1, pressure2_load3_diame1,
Fc2_load3_diame1, Fb2_load3_diame1, nu2_load3_diame1, Fba2_load3_diame1,
Fdyn2_load3_diame1, Fbrake2_load3_diame1, Fra2_load3_diame1,
Fg2_load3_diame1, Fext2_load3_diame1, Fb2_load3_diame1,
Ftotal2_load3_diame1, acc2_load3_diame1, s2_load3_diame1,
time_eq2_load3_diame1,
distance_eq2_load3_diame1, deceleration_eq2_load3_diame1,
velocity2_load3_diame2, time2_load3_diame2, pressure2_load3_diame2,
Fc2_load3_diame2, Fb2_load3_diame2, nu2_load3_diame2, Fba2_load3_diame2,
Fdyn2_load3_diame2, Fbrake2_load3_diame2, Fra2_load3_diame2,
Fg2_load3_diame2, Fext2_load3_diame2, Fb2_load3_diame2,
Ftotal2_load3_diame2, acc2_load3_diame2, s2_load3_diame2,
time_eq2_load3_diame2,
distance_eq2_load3_diame2, deceleration_eq2_load3_diame2,
velocity2_load3_diame3, time2_load3_diame3, pressure2_load3_diame3,
Fc2_load3_diame3, Fb2_load3_diame3, nu2_load3_diame3, Fba2_load3_diame3,
Fdyn2_load3_diame3, Fbrake2_load3_diame3, Fra2_load3_diame3,
Fg2_load3_diame3, Fext2_load3_diame3, Fb2_load3_diame3,
Ftotal2_load3_diame3, acc2_load3_diame3, s2_load3_diame3,
time_eq2_load3_diame3,
distance_eq2_load3_diame3, deceleration_eq2_load3_diame3,
velocity2_load4_diame1, time2_load4_diame1, pressure2_load4_diame1,
Fc2_load4_diame1, Fb2_load4_diame1, nu2_load4_diame1, Fba2_load4_diame1,
Fdyn2_load4_diame1, Fbrake2_load4_diame1, Fra2_load4_diame1,
Fg2_load4_diame1, Fext2_load4_diame1, Fb2_load4_diame1,
Ftotal2_load4_diame1, acc2_load4_diame1, s2_load4_diame1,
time_eq2_load4_diame1,
distance_eq2_load4_diame1, deceleration_eq2_load4_diame1,
velocity2_load4_diame2, time2_load4_diame2, pressure2_load4_diame2,
Fc2_load4_diame2, Fb2_load4_diame2, nu2_load4_diame2, Fba2_load4_diame2,
Fdyn2_load4_diame2, Fbrake2_load4_diame2, Fra2_load4_diame2,
Fg2_load4_diame2, Fext2_load4_diame2, Fb2_load4_diame2,
Ftotal2_load4_diame2, acc2_load4_diame2, s2_load4_diame2,
time_eq2_load4_diame2,
distance_eq2_load4_diame2, deceleration_eq2_load4_diame2,
velocity2_load4_diame3, time2_load4_diame3, pressure2_load4_diame3,
Fc2_load4_diame3, Fb2_load4_diame3, nu2_load4_diame3, Fba2_load4_diame3,

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```

Fdyn2_load4_diame3,      Fbrake2_load4_diame3,      Fra2_load4_diame3,
Fg2_load4_diame3,       Fext2_load4_diame3,       Fb2_load4_diame3,
Ftotal2_load4_diame3,   acc2_load4_diame3,        s2_load4_diame3,
time_eq2_load4_diame3,
distance_eq2_load4_diame3, deceleration_eq2_load4_diame3, velocity2
_load5_diame1,          time2_load5_diame1,        pressure2_load5_diame1,
Fc2_load5_diame1, Fb2_load5_diame1, nu2_load5_diame1, Fba2_load5_diame1,
Fdyn2_load5_diame1,      Fbrake2_load5_diame1,      Fra2_load5_diame1,
Fg2_load5_diame1,       Fext2_load5_diame1,       Fb2_load5_diame1,
Ftotal2_load5_diame1,   acc2_load5_diame1,        s2_load5_diame1,
time_eq2_load5_diame1,
distance_eq2_load5_diame1,          deceleration_eq2_load5_diame1,
velocity2_load5_diame2, time2_load5_diame2, pressure2_load5_diame2,
Fc2_load5_diame2, Fb2_load5_diame2, nu2_load5_diame2, Fba2_load5_diame2,
Fdyn2_load5_diame2,      Fbrake2_load5_diame2,      Fra2_load5_diame2,
Fg2_load5_diame2,       Fext2_load5_diame2,       Fb2_load5_diame2,
Ftotal2_load5_diame2,   acc2_load5_diame2,        s2_load5_diame2,
time_eq2_load5_diame2,
distance_eq2_load5_diame2,          deceleration_eq2_load5_diame2,
velocity2_load5_diame3, time2_load5_diame3, pressure2_load5_diame3,
Fc2_load5_diame3, Fb2_load5_diame3, nu2_load5_diame3, Fba2_load5_diame3,
Fdyn2_load5_diame3,      Fbrake2_load5_diame3,      Fra2_load5_diame3,
Fg2_load5_diame3,       Fext2_load5_diame3,       Fb2_load5_diame3,
Ftotal2_load5_diame3,   acc2_load5_diame3,        s2_load5_diame3,
time_eq2_load5_diame3,
distance_eq2_load5_diame3, deceleration_eq2_load5_diame3, velocity2,
time2, pressure2, Fc2, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7, Fba8,
Fba9, nu2, Fba2, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8,
Fdyn9, Fbrake2, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9,
Fg1, Fg2, Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext2, FB2, Ftotal2, acc2,
s2, time_eq2, distance_eq2, deceleration_eq2] = load_diam(stp, x,
condition, sav, GeneralitiesVehicleComp_Data, NominalMass_Data,
DynamicMass_Data, WheelDiameter_Data, VehicleResistance_Data,
FrictionBrakeForces_Data, DynamicBrakeForces_Data, TrackBrake_Data,
Initial_Data, Degraded_Data_2, Pressure_Vel_Data);

Fba1_cond2 = Fba1;
Fba2_cond2 = Fba2;
Fba3_cond2 = Fba3;
Fba4_cond2 = Fba4;
Fba5_cond2 = Fba5;
Fba6_cond2 = Fba6;
Fba7_cond2 = Fba7;
Fba8_cond2 = Fba8;
Fba9_cond2 = Fba9;

Fdyn1_cond2 = Fdyn1;
Fdyn2_cond2 = Fdyn2;
Fdyn3_cond2 = Fdyn3;
Fdyn4_cond2 = Fdyn4;
Fdyn5_cond2 = Fdyn5;
Fdyn6_cond2 = Fdyn6;
Fdyn7_cond2 = Fdyn7;
Fdyn8_cond2 = Fdyn8;
Fdyn9_cond2 = Fdyn9;

```

```
Fra1_cond2 = Fra1;
Fra2_cond2 = Fra2;
Fra3_cond2 = Fra3;
Fra4_cond2 = Fra4;
Fra5_cond2 = Fra5;
Fra6_cond2 = Fra6;
Fra7_cond2 = Fra7;
Fra8_cond2 = Fra8;
Fra9_cond2 = Fra9;

Fg1_cond2 = Fg1;
Fg2_cond2 = Fg2;
Fg3_cond2 = Fg3;
Fg4_cond2 = Fg4;
Fg5_cond2 = Fg5;
Fg6_cond2 = Fg6;
Fg7_cond2 = Fg7;
Fg8_cond2 = Fg8;
Fg9_cond2 = Fg9;

s_load1_diame1_1 = s2_load1_diame1;
s_load1_diame2_1 = s2_load1_diame2;
s_load1_diame3_1 = s2_load1_diame3;
s_load2_diame1_1 = s2_load2_diame1;
s_load2_diame2_1 = s2_load2_diame2;
s_load2_diame3_1 = s2_load2_diame3;
s_load3_diame1_1 = s2_load3_diame1;
s_load3_diame2_1 = s2_load3_diame2;
s_load3_diame3_1 = s2_load3_diame3;
s_load4_diame1_1 = s2_load4_diame1;
s_load4_diame2_1 = s2_load4_diame2;
s_load4_diame3_1 = s2_load4_diame3;
s_load5_diame1_1 = s2_load5_diame1;
s_load5_diame2_1 = s2_load5_diame2;
s_load5_diame3_1 = s2_load5_diame3;

if length(s_load1_diame1_2) ~= 0
    dev = abs((s_load1_diame1_2(end) -
s_load1_diame1_1(end))/s_load1_diame1_1(end));
    deviation2_load1_diame1(condition) = dev;
end
if length(s_load1_diame2_2) ~= 0
    dev = abs((s_load1_diame2_2(end) -
s_load1_diame2_1(end))/s_load1_diame2_1(end));
    deviation2_load1_diame2(condition) = dev;
end
if length(s_load1_diame3_2) ~= 0
    dev = abs((s_load1_diame3_2(end) -
s_load1_diame3_1(end))/s_load1_diame3_1(end));
    deviation2_load1_diame3(condition) = dev;
end
if length(s_load2_diame1_2) ~= 0
    dev = abs((s_load2_diame1_2(end) -
s_load2_diame1_1(end))/s_load2_diame1_1(end));
    deviation2_load2_diame1(condition) = dev;
end
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario

```
    if length(s_load2_diame2_2) ~= 0
        dev = abs((s_load2_diame2_2(end) -
s_load2_diame2_1(end))/s_load2_diame2_1(end));
        deviation2_load2_diame2(condition) = dev;
    end
    if length(s_load2_diame3_2) ~= 0
        dev = abs((s_load2_diame3_2(end) -
s_load2_diame3_1(end))/s_load2_diame3_1(end));
        deviation2_load2_diame3(condition) = dev;
    end
    if length(s_load3_diame1_2) ~= 0
        dev = abs((s_load3_diame1_2(end) -
s_load3_diame1_1(end))/s_load3_diame1_1(end));
        deviation2_load3_diame1(condition) = dev;
    end
    if length(s_load3_diame2_2) ~= 0
        dev = abs((s_load3_diame2_2(end) -
s_load3_diame2_1(end))/s_load3_diame2_1(end));
        deviation2_load3_diame2(condition) = dev;
    end
    if length(s_load3_diame3_2) ~= 0
        dev = abs((s_load3_diame3_2(end) -
s_load3_diame3_1(end))/s_load3_diame3_1(end));
        deviation2_load3_diame3(condition) = dev;
    end
    if length(s_load4_diame1_2) ~= 0
        dev = abs((s_load4_diame1_2(end) -
s_load4_diame1_1(end))/s_load4_diame1_1(end));
        deviation2_load4_diame1(condition) = dev;
    end
    if length(s_load4_diame2_2) ~= 0
        dev = abs((s_load4_diame2_2(end) -
s_load4_diame2_1(end))/s_load4_diame2_1(end));
        deviation2_load4_diame2(condition) = dev;
    end
    if length(s_load4_diame3_2) ~= 0
        dev = abs((s_load4_diame3_2(end) -
s_load4_diame3_1(end))/s_load4_diame3_1(end));
        deviation2_load4_diame3(condition) = dev;
    end
    if length(s_load5_diame1_2) ~= 0
        dev = abs((s_load5_diame1_2(end) -
s_load5_diame1_1(end))/s_load5_diame1_1(end));
        deviation2_load5_diame1(condition) = dev;
    end
    if length(s_load5_diame2_2) ~= 0
        dev = abs((s_load5_diame2_2(end) -
s_load5_diame2_1(end))/s_load5_diame2_1(end));
        deviation2_load5_diame2(condition) = dev;
    end
    if length(s_load5_diame3_2) ~= 0
        dev = abs((s_load5_diame3_2(end) -
s_load5_diame3_1(end))/s_load5_diame3_1(end));
        deviation2_load5_diame3(condition) = dev;
    end
end
end
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
if condition == 3
    sav = 0;
    [velocity3_load1_diame1,    time3_load1_diame1,
pressure3_load1_diame1,      Fc3_load1_diame1,    Fb3_load1_diame1,
nu3_load1_diame1,          Fba3_load1_diame1,    Fdyn3_load1_diame1,
Fbrake3_load1_diame1,      Fra3_load1_diame1,    Fg3_load1_diame1,
Fext3_load1_diame1,        Fb3_load1_diame1,      Ftotal3_load1_diame1,
acc3_load1_diame1,         s3_load1_diame1,      time_eq3_load1_diame1,
distance_eq3_load1_diame1,    deceleration_eq3_load1_diame1,
velocity3_load1_diame2,    time3_load1_diame2,    pressure3_load1_diame2,
Fc3_load1_diame2, Fb3_load1_diame2, nu3_load1_diame2, Fba3_load1_diame2,
Fdyn3_load1_diame2, Fbrake3_load1_diame2, Fra3_load1_diame2,
Fg3_load1_diame2, Fext3_load1_diame2, Fb3_load1_diame2,
Ftotal3_load1_diame2, acc3_load1_diame2, s3_load1_diame2,
time_eq3_load1_diame2,
distance_eq3_load1_diame2,    deceleration_eq3_load1_diame2,
velocity3_load1_diame3,    time3_load1_diame3,    pressure3_load1_diame3,
Fc3_load1_diame3, Fb3_load1_diame3, nu3_load1_diame3, Fba3_load1_diame3,
Fdyn3_load1_diame3, Fbrake3_load1_diame3, Fra3_load1_diame3,
Fg3_load1_diame3, Fext3_load1_diame3, Fb3_load1_diame3,
Ftotal3_load1_diame3, acc3_load1_diame3, s3_load1_diame3,
time_eq3_load1_diame3,
distance_eq3_load1_diame3,    deceleration_eq3_load1_diame3,
velocity3_load2_diame1,    time3_load2_diame1,    pressure3_load2_diame1,
Fc3_load2_diame1, Fb3_load2_diame1, nu3_load2_diame1, Fba3_load2_diame1,
Fdyn3_load2_diame1, Fbrake3_load2_diame1, Fra3_load2_diame1,
Fg3_load2_diame1, Fext3_load2_diame1, Fb3_load2_diame1,
Ftotal3_load2_diame1, acc3_load2_diame1, s3_load2_diame1,
time_eq3_load2_diame1,
distance_eq3_load2_diame1,    deceleration_eq3_load2_diame1,
velocity3_load2_diame2,    time3_load2_diame2,    pressure3_load2_diame2,
Fc3_load2_diame2, Fb3_load2_diame2, nu3_load2_diame2, Fba3_load2_diame2,
Fdyn3_load2_diame2, Fbrake3_load2_diame2, Fra3_load2_diame2,
Fg3_load2_diame2, Fext3_load2_diame2, Fb3_load2_diame2,
Ftotal3_load2_diame2, acc3_load2_diame2, s3_load2_diame2,
time_eq3_load2_diame2,
distance_eq3_load2_diame2,    deceleration_eq3_load2_diame2,
velocity3_load2_diame3,    time3_load2_diame3,    pressure3_load2_diame3,
Fc3_load2_diame3, Fb3_load2_diame3, nu3_load2_diame3, Fba3_load2_diame3,
Fdyn3_load2_diame3, Fbrake3_load2_diame3, Fra3_load2_diame3,
Fg3_load2_diame3, Fext3_load2_diame3, Fb3_load2_diame3,
Ftotal3_load2_diame3, acc3_load2_diame3, s3_load2_diame3,
time_eq3_load2_diame3,
distance_eq3_load2_diame3,    deceleration_eq3_load2_diame3,
velocity3_load3_diame1,    time3_load3_diame1,    pressure3_load3_diame1,
Fc3_load3_diame1, Fb3_load3_diame1, nu3_load3_diame1, Fba3_load3_diame1,
Fdyn3_load3_diame1, Fbrake3_load3_diame1, Fra3_load3_diame1,
Fg3_load3_diame1, Fext3_load3_diame1, Fb3_load3_diame1,
Ftotal3_load3_diame1, acc3_load3_diame1, s3_load3_diame1,
time_eq3_load3_diame1,
distance_eq3_load3_diame1,    deceleration_eq3_load3_diame1,
velocity3_load3_diame2,    time3_load3_diame2,    pressure3_load3_diame2,
Fc3_load3_diame2, Fb3_load3_diame2, nu3_load3_diame2, Fba3_load3_diame2,
Fdyn3_load3_diame2, Fbrake3_load3_diame2, Fra3_load3_diame2,
Fg3_load3_diame2, Fext3_load3_diame2, Fb3_load3_diame2,
Ftotal3_load3_diame2, acc3_load3_diame2, s3_load3_diame2,
```


Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
time_eq3_load3_diame2,
distance_eq3_load3_diame2,                deceleration_eq3_load3_diame2,
velocity3_load3_diame3,  time3_load3_diame3,  pressure3_load3_diame3,
Fc3_load3_diame3, Fb3_load3_diame3, nu3_load3_diame3, Fba3_load3_diame3,
Fdyn3_load3_diame3,  Fbrake3_load3_diame3,  Fra3_load3_diame3,
Fg3_load3_diame3,  Fext3_load3_diame3,  Fb3_load3_diame3,
Ftotal3_load3_diame3,  acc3_load3_diame3,  s3_load3_diame3,
time_eq3_load3_diame3,
distance_eq3_load3_diame3,                deceleration_eq3_load3_diame3,
velocity3_load4_diame1,  time3_load4_diame1,  pressure3_load4_diame1,
Fc3_load4_diame1, Fb3_load4_diame1, nu3_load4_diame1, Fba3_load4_diame1,
Fdyn3_load4_diame1,  Fbrake3_load4_diame1,  Fra3_load4_diame1,
Fg3_load4_diame1,  Fext3_load4_diame1,  Fb3_load4_diame1,
Ftotal3_load4_diame1,  acc3_load4_diame1,  s3_load4_diame1,
time_eq3_load4_diame1,
distance_eq3_load4_diame1,                deceleration_eq3_load4_diame1,
velocity3_load4_diame2,  time3_load4_diame2,  pressure3_load4_diame2,
Fc3_load4_diame2, Fb3_load4_diame2, nu3_load4_diame2, Fba3_load4_diame2,
Fdyn3_load4_diame2,  Fbrake3_load4_diame2,  Fra3_load4_diame2,
Fg3_load4_diame2,  Fext3_load4_diame2,  Fb3_load4_diame2,
Ftotal3_load4_diame2,  acc3_load4_diame2,  s3_load4_diame2,
time_eq3_load4_diame2,
distance_eq3_load4_diame2,                deceleration_eq3_load4_diame2,
velocity3_load4_diame3,  time3_load4_diame3,  pressure3_load4_diame3,
Fc3_load4_diame3, Fb3_load4_diame3, nu3_load4_diame3, Fba3_load4_diame3,
Fdyn3_load4_diame3,  Fbrake3_load4_diame3,  Fra3_load4_diame3,
Fg3_load4_diame3,  Fext3_load4_diame3,  Fb3_load4_diame3,
Ftotal3_load4_diame3,  acc3_load4_diame3,  s3_load4_diame3,
time_eq3_load4_diame3,
distance_eq3_load4_diame3,  deceleration_eq3_load4_diame3,  velocity3
_load5_diame1,  time3_load5_diame1,  pressure3_load5_diame1,
Fc3_load5_diame1, Fb3_load5_diame1, nu3_load5_diame1, Fba3_load5_diame1,
Fdyn3_load5_diame1,  Fbrake3_load5_diame1,  Fra3_load5_diame1,
Fg3_load5_diame1,  Fext3_load5_diame1,  Fb3_load5_diame1,
Ftotal3_load5_diame1,  acc3_load5_diame1,  s3_load5_diame1,
time_eq3_load5_diame1,
distance_eq3_load5_diame1,                deceleration_eq3_load5_diame1,
velocity3_load5_diame2,  time3_load5_diame2,  pressure3_load5_diame2,
Fc3_load5_diame2, Fb3_load5_diame2, nu3_load5_diame2, Fba3_load5_diame2,
Fdyn3_load5_diame2,  Fbrake3_load5_diame2,  Fra3_load5_diame2,
Fg3_load5_diame2,  Fext3_load5_diame2,  Fb3_load5_diame2,
Ftotal3_load5_diame2,  acc3_load5_diame2,  s3_load5_diame2,
time_eq3_load5_diame2,
distance_eq3_load5_diame2,                deceleration_eq3_load5_diame2,
velocity3_load5_diame3,  time3_load5_diame3,  pressure3_load5_diame3,
Fc3_load5_diame3, Fb3_load5_diame3, nu3_load5_diame3, Fba3_load5_diame3,
Fdyn3_load5_diame3,  Fbrake3_load5_diame3,  Fra3_load5_diame3,
Fg3_load5_diame3,  Fext3_load5_diame3,  Fb3_load5_diame3,
Ftotal3_load5_diame3,  acc3_load5_diame3,  s3_load5_diame3,
time_eq3_load5_diame3,
distance_eq3_load5_diame3,  deceleration_eq3_load5_diame3,  velocity3,
time3, pressure3, Fc3, Fb3, nu3, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake3, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9, Fg1, Fg2,
Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext3, FB3, Ftotal3, acc3, s3,
time_eq3,  distance_eq3,  deceleration_eq3] = load_diam(stp2, x,
```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```

condition, sav, GeneralitiesVehicleComp_Data, NominalMass_Data,
DynamicMass_Data, WheelDiameter_Data, VehicleResistance_Data,
FrictionBrakeForces_Data, DynamicBrakeForces_Data, TrackBrake_Data,
Initial_Data, Degraded_Data_3, Pressure_Vel_Data);

s_load1_diame1_2 = s3_load1_diame1;
s_load1_diame2_2 = s3_load1_diame2;
s_load1_diame3_2 = s3_load1_diame3;
s_load2_diame1_2 = s3_load2_diame1;
s_load2_diame2_2 = s3_load2_diame2;
s_load2_diame3_2 = s3_load2_diame3;
s_load3_diame1_2 = s3_load3_diame1;
s_load3_diame2_2 = s3_load3_diame2;
s_load3_diame3_2 = s3_load3_diame3;
s_load4_diame1_2 = s3_load4_diame1;
s_load4_diame2_2 = s3_load4_diame2;
s_load4_diame3_2 = s3_load4_diame3;
s_load5_diame1_2 = s3_load5_diame1;
s_load5_diame2_2 = s3_load5_diame2;
s_load5_diame3_2 = s3_load5_diame3;

sav = 1;

[velocity3_load1_diame1, time3_load1_diame1,
pressure3_load1_diame1, Fc3_load1_diame1, Fb3_load1_diame1,
nu3_load1_diame1, Fba3_load1_diame1, Fdyn3_load1_diame1,
Fbrake3_load1_diame1, Fra3_load1_diame1, Fg3_load1_diame1,
Fext3_load1_diame1, Fb3_load1_diame1, Ftotal3_load1_diame1,
acc3_load1_diame1, s3_load1_diame1, time_eq3_load1_diame1,
distance_eq3_load1_diame1, deceleration_eq3_load1_diame1,
velocity3_load1_diame2, time3_load1_diame2, pressure3_load1_diame2,
Fc3_load1_diame2, Fb3_load1_diame2, nu3_load1_diame2, Fba3_load1_diame2,
Fdyn3_load1_diame2, Fbrake3_load1_diame2, Fra3_load1_diame2,
Fg3_load1_diame2, Fext3_load1_diame2, Fb3_load1_diame2,
Ftotal3_load1_diame2, acc3_load1_diame2, s3_load1_diame2,
time_eq3_load1_diame2,
distance_eq3_load1_diame2, deceleration_eq3_load1_diame2,
velocity3_load1_diame3, time3_load1_diame3, pressure3_load1_diame3,
Fc3_load1_diame3, Fb3_load1_diame3, nu3_load1_diame3, Fba3_load1_diame3,
Fdyn3_load1_diame3, Fbrake3_load1_diame3, Fra3_load1_diame3,
Fg3_load1_diame3, Fext3_load1_diame3, Fb3_load1_diame3,
Ftotal3_load1_diame3, acc3_load1_diame3, s3_load1_diame3,
time_eq3_load1_diame3,
distance_eq3_load1_diame3, deceleration_eq3_load1_diame3,
velocity3_load2_diame1, time3_load2_diame1, pressure3_load2_diame1,
Fc3_load2_diame1, Fb3_load2_diame1, nu3_load2_diame1, Fba3_load2_diame1,
Fdyn3_load2_diame1, Fbrake3_load2_diame1, Fra3_load2_diame1,
Fg3_load2_diame1, Fext3_load2_diame1, Fb3_load2_diame1,
Ftotal3_load2_diame1, acc3_load2_diame1, s3_load2_diame1,
time_eq3_load2_diame1,
distance_eq3_load2_diame1, deceleration_eq3_load2_diame1,
velocity3_load2_diame2, time3_load2_diame2, pressure3_load2_diame2,
Fc3_load2_diame2, Fb3_load2_diame2, nu3_load2_diame2, Fba3_load2_diame2,
Fdyn3_load2_diame2, Fbrake3_load2_diame2, Fra3_load2_diame2,
Fg3_load2_diame2, Fext3_load2_diame2, Fb3_load2_diame2,
Ftotal3_load2_diame2, acc3_load2_diame2, s3_load2_diame2,
time_eq3_load2_diame2,

```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
distance_eq3_load2_diame2,                deceleration_eq3_load2_diame2,
velocity3_load2_diame3,    time3_load2_diame3,    pressure3_load2_diame3,
Fc3_load2_diame3,    Fb3_load2_diame3,    nu3_load2_diame3,    Fba3_load2_diame3,
Fdyn3_load2_diame3,    Fbrake3_load2_diame3,    Fra3_load2_diame3,
Fg3_load2_diame3,    Fext3_load2_diame3,    Fb3_load2_diame3,
Ftotal3_load2_diame3,    acc3_load2_diame3,    s3_load2_diame3,
time_eq3_load2_diame3,
distance_eq3_load2_diame3,                deceleration_eq3_load2_diame3,
velocity3_load3_diame1,    time3_load3_diame1,    pressure3_load3_diame1,
Fc3_load3_diame1,    Fb3_load3_diame1,    nu3_load3_diame1,    Fba3_load3_diame1,
Fdyn3_load3_diame1,    Fbrake3_load3_diame1,    Fra3_load3_diame1,
Fg3_load3_diame1,    Fext3_load3_diame1,    Fb3_load3_diame1,
Ftotal3_load3_diame1,    acc3_load3_diame1,    s3_load3_diame1,
time_eq3_load3_diame1,
distance_eq3_load3_diame1,                deceleration_eq3_load3_diame1,
velocity3_load3_diame2,    time3_load3_diame2,    pressure3_load3_diame2,
Fc3_load3_diame2,    Fb3_load3_diame2,    nu3_load3_diame2,    Fba3_load3_diame2,
Fdyn3_load3_diame2,    Fbrake3_load3_diame2,    Fra3_load3_diame2,
Fg3_load3_diame2,    Fext3_load3_diame2,    Fb3_load3_diame2,
Ftotal3_load3_diame2,    acc3_load3_diame2,    s3_load3_diame2,
time_eq3_load3_diame2,
distance_eq3_load3_diame2,                deceleration_eq3_load3_diame2,
velocity3_load3_diame3,    time3_load3_diame3,    pressure3_load3_diame3,
Fc3_load3_diame3,    Fb3_load3_diame3,    nu3_load3_diame3,    Fba3_load3_diame3,
Fdyn3_load3_diame3,    Fbrake3_load3_diame3,    Fra3_load3_diame3,
Fg3_load3_diame3,    Fext3_load3_diame3,    Fb3_load3_diame3,
Ftotal3_load3_diame3,    acc3_load3_diame3,    s3_load3_diame3,
time_eq3_load3_diame3,
distance_eq3_load3_diame3,                deceleration_eq3_load3_diame3,
velocity3_load4_diame1,    time3_load4_diame1,    pressure3_load4_diame1,
Fc3_load4_diame1,    Fb3_load4_diame1,    nu3_load4_diame1,    Fba3_load4_diame1,
Fdyn3_load4_diame1,    Fbrake3_load4_diame1,    Fra3_load4_diame1,
Fg3_load4_diame1,    Fext3_load4_diame1,    Fb3_load4_diame1,
Ftotal3_load4_diame1,    acc3_load4_diame1,    s3_load4_diame1,
time_eq3_load4_diame1,
distance_eq3_load4_diame1,                deceleration_eq3_load4_diame1,
velocity3_load4_diame2,    time3_load4_diame2,    pressure3_load4_diame2,
Fc3_load4_diame2,    Fb3_load4_diame2,    nu3_load4_diame2,    Fba3_load4_diame2,
Fdyn3_load4_diame2,    Fbrake3_load4_diame2,    Fra3_load4_diame2,
Fg3_load4_diame2,    Fext3_load4_diame2,    Fb3_load4_diame2,
Ftotal3_load4_diame2,    acc3_load4_diame2,    s3_load4_diame2,
time_eq3_load4_diame2,
distance_eq3_load4_diame2,                deceleration_eq3_load4_diame2,
velocity3_load4_diame3,    time3_load4_diame3,    pressure3_load4_diame3,
Fc3_load4_diame3,    Fb3_load4_diame3,    nu3_load4_diame3,    Fba3_load4_diame3,
Fdyn3_load4_diame3,    Fbrake3_load4_diame3,    Fra3_load4_diame3,
Fg3_load4_diame3,    Fext3_load4_diame3,    Fb3_load4_diame3,
Ftotal3_load4_diame3,    acc3_load4_diame3,    s3_load4_diame3,
time_eq3_load4_diame3,
distance_eq3_load4_diame3,    deceleration_eq3_load4_diame3,    velocity3
_load5_diame1,    time3_load5_diame1,    pressure3_load5_diame1,
Fc3_load5_diame1,    Fb3_load5_diame1,    nu3_load5_diame1,    Fba3_load5_diame1,
Fdyn3_load5_diame1,    Fbrake3_load5_diame1,    Fra3_load5_diame1,
Fg3_load5_diame1,    Fext3_load5_diame1,    Fb3_load5_diame1,
Ftotal3_load5_diame1,    acc3_load5_diame1,    s3_load5_diame1,
time_eq3_load5_diame1,
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
distance_eq3_load5_diame1,          deceleration_eq3_load5_diame1,
velocity3_load5_diame2,    time3_load5_diame2,    pressure3_load5_diame2,
Fc3_load5_diame2, Fb3_load5_diame2, nu3_load5_diame2, Fba3_load5_diame2,
Fdyn3_load5_diame2,    Fbrake3_load5_diame2,    Fra3_load5_diame2,
Fg3_load5_diame2,    Fext3_load5_diame2,    Fb3_load5_diame2,
Ftotal3_load5_diame2,    acc3_load5_diame2,    s3_load5_diame2,
time_eq3_load5_diame2,
distance_eq3_load5_diame2,          deceleration_eq3_load5_diame2,
velocity3_load5_diame3,    time3_load5_diame3,    pressure3_load5_diame3,
Fc3_load5_diame3, Fb3_load5_diame3, nu3_load5_diame3, Fba3_load5_diame3,
Fdyn3_load5_diame3,    Fbrake3_load5_diame3,    Fra3_load5_diame3,
Fg3_load5_diame3,    Fext3_load5_diame3,    Fb3_load5_diame3,
Ftotal3_load5_diame3,    acc3_load5_diame3,    s3_load5_diame3,
time_eq3_load5_diame3,
distance_eq3_load5_diame3,    deceleration_eq3_load5_diame3,    velocity3,
time3, pressure3, Fc3, Fb3, nu3, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake3, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9, Fg1, Fg2,
Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext3, FB3, Ftotal3, acc3, s3,
time_eq3, distance_eq3, deceleration_eq3] = load_diam(stp, x, condition,
sav, GeneralitiesVehicleComp_Data, NominalMass_Data, DynamicMass_Data,
WheelDiameter_Data, VehicleResistance_Data, FrictionBrakeForces_Data,
DynamicBrakeForces_Data, TrackBrake_Data, Initial_Data, Degraded_Data_3,
Pressure_Vel_Data);
```

```
Fba1_cond3 = Fba1;
Fba2_cond3 = Fba2;
Fba3_cond3 = Fba3;
Fba4_cond3 = Fba4;
Fba5_cond3 = Fba5;
Fba6_cond3 = Fba6;
Fba7_cond3 = Fba7;
Fba8_cond3 = Fba8;
Fba9_cond3 = Fba9;
```

```
Fdyn1_cond3 = Fdyn1;
Fdyn2_cond3 = Fdyn2;
Fdyn3_cond3 = Fdyn3;
Fdyn4_cond3 = Fdyn4;
Fdyn5_cond3 = Fdyn5;
Fdyn6_cond3 = Fdyn6;
Fdyn7_cond3 = Fdyn7;
Fdyn8_cond3 = Fdyn8;
Fdyn9_cond3 = Fdyn9;
```

```
Fra1_cond3 = Fra1;
Fra2_cond3 = Fra2;
Fra3_cond3 = Fra3;
Fra4_cond3 = Fra4;
Fra5_cond3 = Fra5;
Fra6_cond3 = Fra6;
Fra7_cond3 = Fra7;
Fra8_cond3 = Fra8;
Fra9_cond3 = Fra9;
```

```
Fg1_cond3 = Fg1;
```

```
Fg2_cond3 = Fg2;
Fg3_cond3 = Fg3;
Fg4_cond3 = Fg4;
Fg5_cond3 = Fg5;
Fg6_cond3 = Fg6;
Fg7_cond3 = Fg7;
Fg8_cond3 = Fg8;
Fg9_cond3 = Fg9;

s_load1_diame1_1 = s3_load1_diame1;
s_load1_diame2_1 = s3_load1_diame2;
s_load1_diame3_1 = s3_load1_diame3;
s_load2_diame1_1 = s3_load2_diame1;
s_load2_diame2_1 = s3_load2_diame2;
s_load2_diame3_1 = s3_load2_diame3;
s_load3_diame1_1 = s3_load3_diame1;
s_load3_diame2_1 = s3_load3_diame2;
s_load3_diame3_1 = s3_load3_diame3;
s_load4_diame1_1 = s3_load4_diame1;
s_load4_diame2_1 = s3_load4_diame2;
s_load4_diame3_1 = s3_load4_diame3;
s_load5_diame1_1 = s3_load5_diame1;
s_load5_diame2_1 = s3_load5_diame2;
s_load5_diame3_1 = s3_load5_diame3;

if length(s_load1_diame1_2) ~= 0
    dev = abs((s_load1_diame1_2(end) -
s_load1_diame1_1(end))/s_load1_diame1_1(end));
    deviation3_load1_diame1(condition) = dev;
end
if length(s_load1_diame2_2) ~= 0
    dev = abs((s_load1_diame2_2(end) -
s_load1_diame2_1(end))/s_load1_diame2_1(end));
    deviation3_load1_diame2(condition) = dev;
end
if length(s_load1_diame3_2) ~= 0
    dev = abs((s_load1_diame3_2(end) -
s_load1_diame3_1(end))/s_load1_diame3_1(end));
    deviation3_load1_diame3(condition) = dev;
end
if length(s_load2_diame1_2) ~= 0
    dev = abs((s_load2_diame1_2(end) -
s_load2_diame1_1(end))/s_load2_diame1_1(end));
    deviation3_load2_diame1(condition) = dev;
end
if length(s_load2_diame2_2) ~= 0
    dev = abs((s_load2_diame2_2(end) -
s_load2_diame2_1(end))/s_load2_diame2_1(end));
    deviation3_load2_diame2(condition) = dev;
end
if length(s_load2_diame3_2) ~= 0
    dev = abs((s_load2_diame3_2(end) -
s_load2_diame3_1(end))/s_load2_diame3_1(end));
    deviation3_load2_diame3(condition) = dev;
end
if length(s_load3_diame1_2) ~= 0
```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```

                                dev      =      abs((s_load3_diame1_2(end) -
s_load3_diame1_1(end))/s_load3_diame1_1(end));
                                deviation3_load3_diame1(condition) = dev;
                                end
                                if length(s_load3_diame2_2) ~= 0
                                        dev      =      abs((s_load3_diame2_2(end) -
s_load3_diame2_1(end))/s_load3_diame2_1(end));
                                        deviation3_load3_diame2(condition) = dev;
                                end
                                if length(s_load3_diame3_2) ~= 0
                                        dev      =      abs((s_load3_diame3_2(end) -
s_load3_diame3_1(end))/s_load3_diame3_1(end));
                                        deviation3_load3_diame3(condition) = dev;
                                end
                                if length(s_load4_diame1_2) ~= 0
                                        dev      =      abs((s_load4_diame1_2(end) -
s_load4_diame1_1(end))/s_load4_diame1_1(end));
                                        deviation3_load4_diame1(condition) = dev;
                                end
                                if length(s_load4_diame2_2) ~= 0
                                        dev      =      abs((s_load4_diame2_2(end) -
s_load4_diame2_1(end))/s_load4_diame2_1(end));
                                        deviation3_load4_diame2(condition) = dev;
                                end
                                if length(s_load4_diame3_2) ~= 0
                                        dev      =      abs((s_load4_diame3_2(end) -
s_load4_diame3_1(end))/s_load4_diame3_1(end));
                                        deviation3_load4_diame3(condition) = dev;
                                end
                                if length(s_load5_diame1_2) ~= 0
                                        dev      =      abs((s_load5_diame1_2(end) -
s_load5_diame1_1(end))/s_load5_diame1_1(end));
                                        deviation3_load5_diame1(condition) = dev;
                                end
                                if length(s_load5_diame2_2) ~= 0
                                        dev      =      abs((s_load5_diame2_2(end) -
s_load5_diame2_1(end))/s_load5_diame2_1(end));
                                        deviation3_load5_diame2(condition) = dev;
                                end
                                if length(s_load5_diame3_2) ~= 0
                                        dev      =      abs((s_load5_diame3_2(end) -
s_load5_diame3_1(end))/s_load5_diame3_1(end));
                                        deviation3_load5_diame3(condition) = dev;
                                end
                                end
                                if condition == 4
                                        sav = 0;

                                [velocity4_load1_diame1,      time4_load1_diame1,
pressure4_load1_diame1,      Fc4_load1_diame1,      Fb4_load1_diame1,
nu4_load1_diame1,      Fba4_load1_diame1,      Fdyn4_load1_diame1,
Fbrake4_load1_diame1,      Fra4_load1_diame1,      Fg4_load1_diame1,
Fext4_load1_diame1,      Fb4_load1_diame1,      Ftotal4_load1_diame1,
acc4_load1_diame1,      s4_load1_diame1,      time_eq4_load1_diame1,
distance_eq4_load1_diame1,      deceleration_eq4_load1_diame1,
velocity4_load1_diame2,      time4_load1_diame2,      pressure4_load1_diame2,
Fc4_load1_diame2,      Fb4_load1_diame2,      nu4_load1_diame2,      Fba4_load1_diame2,

```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

Fdyn4_load1_diame2, Fbrake4_load1_diame2, Fra4_load1_diame2,
Fg4_load1_diame2, Fext4_load1_diame2, Fb4_load1_diame2,
Ftotal4_load1_diame2, acc4_load1_diame2, s4_load1_diame2,
time_eq4_load1_diame2,
distance_eq4_load1_diame2, deceleration_eq4_load1_diame2,
velocity4_load1_diame3, time4_load1_diame3, pressure4_load1_diame3,
Fc4_load1_diame3, Fb4_load1_diame3, nu4_load1_diame3, Fba4_load1_diame3,
Fdyn4_load1_diame3, Fbrake4_load1_diame3, Fra4_load1_diame3,
Fg4_load1_diame3, Fext4_load1_diame3, Fb4_load1_diame3,
Ftotal4_load1_diame3, acc4_load1_diame3, s4_load1_diame3,
time_eq4_load1_diame3,
distance_eq4_load1_diame3, deceleration_eq4_load1_diame3,
velocity4_load2_diame1, time4_load2_diame1, pressure4_load2_diame1,
Fc4_load2_diame1, Fb4_load2_diame1, nu4_load2_diame1, Fba4_load2_diame1,
Fdyn4_load2_diame1, Fbrake4_load2_diame1, Fra4_load2_diame1,
Fg4_load2_diame1, Fext4_load2_diame1, Fb4_load2_diame1,
Ftotal4_load2_diame1, acc4_load2_diame1, s4_load2_diame1,
time_eq4_load2_diame1,
distance_eq4_load2_diame1, deceleration_eq4_load2_diame1,
velocity4_load2_diame2, time4_load2_diame2, pressure4_load2_diame2,
Fc4_load2_diame2, Fb4_load2_diame2, nu4_load2_diame2, Fba4_load2_diame2,
Fdyn4_load2_diame2, Fbrake4_load2_diame2, Fra4_load2_diame2,
Fg4_load2_diame2, Fext4_load2_diame2, Fb4_load2_diame2,
Ftotal4_load2_diame2, acc4_load2_diame2, s4_load2_diame2,
time_eq4_load2_diame2,
distance_eq4_load2_diame2, deceleration_eq4_load2_diame2,
velocity4_load2_diame3, time4_load2_diame3, pressure4_load2_diame3,
Fc4_load2_diame3, Fb4_load2_diame3, nu4_load2_diame3, Fba4_load2_diame3,
Fdyn4_load2_diame3, Fbrake4_load2_diame3, Fra4_load2_diame3,
Fg4_load2_diame3, Fext4_load2_diame3, Fb4_load2_diame3,
Ftotal4_load2_diame3, acc4_load2_diame3, s4_load2_diame3,
time_eq4_load2_diame3,
distance_eq4_load2_diame3, deceleration_eq4_load2_diame3,
velocity4_load3_diame1, time4_load3_diame1, pressure4_load3_diame1,
Fc4_load3_diame1, Fb4_load3_diame1, nu4_load3_diame1, Fba4_load3_diame1,
Fdyn4_load3_diame1, Fbrake4_load3_diame1, Fra4_load3_diame1,
Fg4_load3_diame1, Fext4_load3_diame1, Fb4_load3_diame1,
Ftotal4_load3_diame1, acc4_load3_diame1, s4_load3_diame1,
time_eq4_load3_diame1,
distance_eq4_load3_diame1, deceleration_eq4_load3_diame1,
velocity4_load3_diame2, time4_load3_diame2, pressure4_load3_diame2,
Fc4_load3_diame2, Fb4_load3_diame2, nu4_load3_diame2, Fba4_load3_diame2,
Fdyn4_load3_diame2, Fbrake4_load3_diame2, Fra4_load3_diame2,
Fg4_load3_diame2, Fext4_load3_diame2, Fb4_load3_diame2,
Ftotal4_load3_diame2, acc4_load3_diame2, s4_load3_diame2,
time_eq4_load3_diame2,
distance_eq4_load3_diame2, deceleration_eq4_load3_diame2,
velocity4_load3_diame3, time4_load3_diame3, pressure4_load3_diame3,
Fc4_load3_diame3, Fb4_load3_diame3, nu4_load3_diame3, Fba4_load3_diame3,
Fdyn4_load3_diame3, Fbrake4_load3_diame3, Fra4_load3_diame3,
Fg4_load3_diame3, Fext4_load3_diame3, Fb4_load3_diame3,
Ftotal4_load3_diame3, acc4_load3_diame3, s4_load3_diame3,
time_eq4_load3_diame3,
distance_eq4_load3_diame3, deceleration_eq4_load3_diame3,
velocity4_load4_diame1, time4_load4_diame1, pressure4_load4_diame1,
Fc4_load4_diame1, Fb4_load4_diame1, nu4_load4_diame1, Fba4_load4_diame1,

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```

Fdyn4_load4_diame1,      Fbrake4_load4_diame1,      Fra4_load4_diame1,
Fg4_load4_diame1,       Fext4_load4_diame1,       Fb4_load4_diame1,
Ftotal4_load4_diame1,   acc4_load4_diame1,        s4_load4_diame1,
time_eq4_load4_diame1,
distance_eq4_load4_diame1,      deceleration_eq4_load4_diame1,
velocity4_load4_diame2,   time4_load4_diame2,      pressure4_load4_diame2,
Fc4_load4_diame2, Fb4_load4_diame2, nu4_load4_diame2, Fba4_load4_diame2,
Fdyn4_load4_diame2,      Fbrake4_load4_diame2,      Fra4_load4_diame2,
Fg4_load4_diame2,       Fext4_load4_diame2,       Fb4_load4_diame2,
Ftotal4_load4_diame2,   acc4_load4_diame2,        s4_load4_diame2,
time_eq4_load4_diame2,
distance_eq4_load4_diame2,      deceleration_eq4_load4_diame2,
velocity4_load4_diame3,   time4_load4_diame3,      pressure4_load4_diame3,
Fc4_load4_diame3, Fb4_load4_diame3, nu4_load4_diame3, Fba4_load4_diame3,
Fdyn4_load4_diame3,      Fbrake4_load4_diame3,      Fra4_load4_diame3,
Fg4_load4_diame3,       Fext4_load4_diame3,       Fb4_load4_diame3,
Ftotal4_load4_diame3,   acc4_load4_diame3,        s4_load4_diame3,
time_eq4_load4_diame3,
distance_eq4_load4_diame3,   deceleration_eq4_load4_diame3,   velocity4
_load5_diame1,      time4_load5_diame1,      pressure4_load5_diame1,
Fc4_load5_diame1, Fb4_load5_diame1, nu4_load5_diame1, Fba4_load5_diame1,
Fdyn4_load5_diame1,      Fbrake4_load5_diame1,      Fra4_load5_diame1,
Fg4_load5_diame1,       Fext4_load5_diame1,       Fb4_load5_diame1,
Ftotal4_load5_diame1,   acc4_load5_diame1,        s4_load5_diame1,
time_eq4_load5_diame1,
distance_eq4_load5_diame1,      deceleration_eq4_load5_diame1,
velocity4_load5_diame2,   time4_load5_diame2,      pressure4_load5_diame2,
Fc4_load5_diame2, Fb4_load5_diame2, nu4_load5_diame2, Fba4_load5_diame2,
Fdyn4_load5_diame2,      Fbrake4_load5_diame2,      Fra4_load5_diame2,
Fg4_load5_diame2,       Fext4_load5_diame2,       Fb4_load5_diame2,
Ftotal4_load5_diame2,   acc4_load5_diame2,        s4_load5_diame2,
time_eq4_load5_diame2,
distance_eq4_load5_diame2,      deceleration_eq4_load5_diame2,
velocity4_load5_diame3,   time4_load5_diame3,      pressure4_load5_diame3,
Fc4_load5_diame3, Fb4_load5_diame3, nu4_load5_diame3, Fba4_load5_diame3,
Fdyn4_load5_diame3,      Fbrake4_load5_diame3,      Fra4_load5_diame3,
Fg4_load5_diame3,       Fext4_load5_diame3,       Fb4_load5_diame3,
Ftotal4_load5_diame3,   acc4_load5_diame3,        s4_load5_diame3,
time_eq4_load5_diame3,
distance_eq4_load5_diame3,   deceleration_eq4_load5_diame3,   velocity4,
time4, pressure4, Fc4, Fb4, nu4, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake4, Fra4, Fg1, Fg2, Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext4, FB4,
Ftotal4, acc4, s4, time_eq4, distance_eq4, deceleration_eq4] =
load_diam(stp2, x, condition, sav, GeneralitiesVehicleComp_Data,
NominalMass_Data,      DynamicMass_Data,      WheelDiameter_Data,
VehicleResistance_Data,      FrictionBrakeForces_Data,
DynamicBrakeForces_Data, TrackBrake_Data, Initial_Data, Degraded_Data_4,
Pressure_Vel_Data);

s_load1_diame1_2 = s4_load1_diame1;
s_load1_diame2_2 = s4_load1_diame2;
s_load1_diame3_2 = s4_load1_diame3;
s_load2_diame1_2 = s4_load2_diame1;
s_load2_diame2_2 = s4_load2_diame2;
s_load2_diame3_2 = s4_load2_diame3;

```


Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
s_load3_diame1_2 = s4_load3_diame1;
s_load3_diame2_2 = s4_load3_diame2;
s_load3_diame3_2 = s4_load3_diame3;
s_load4_diame1_2 = s4_load4_diame1;
s_load4_diame2_2 = s4_load4_diame2;
s_load4_diame3_2 = s4_load4_diame3;
s_load5_diame1_2 = s4_load5_diame1;
s_load5_diame2_2 = s4_load5_diame2;
s_load5_diame3_2 = s4_load5_diame3;

sav = 1;

[velocity4_load1_diame1,      time4_load1_diame1,
pressure4_load1_diame1,      Fc4_load1_diame1,      Fb4_load1_diame1,
nu4_load1_diame1,           Fba4_load1_diame1,     Fdyn4_load1_diame1,
Fbrake4_load1_diame1,      Fra4_load1_diame1,     Fg4_load1_diame1,
Fext4_load1_diame1,        Fb4_load1_diame1,     Fttotal4_load1_diame1,
acc4_load1_diame1,         s4_load1_diame1,      time_eq4_load1_diame1,
distance_eq4_load1_diame1,  deceleration_eq4_load1_diame1,
velocity4_load1_diame2,     time4_load1_diame2,   pressure4_load1_diame2,
Fc4_load1_diame2,          Fb4_load1_diame2,     nu4_load1_diame2,      Fba4_load1_diame2,
Fdyn4_load1_diame2,        Fbrake4_load1_diame2, Fra4_load1_diame2,
Fg4_load1_diame2,          Fext4_load1_diame2,   Fb4_load1_diame2,
Fttotal4_load1_diame2,     acc4_load1_diame2,    s4_load1_diame2,
time_eq4_load1_diame2,
distance_eq4_load1_diame2,  deceleration_eq4_load1_diame2,
velocity4_load1_diame3,     time4_load1_diame3,   pressure4_load1_diame3,
Fc4_load1_diame3,          Fb4_load1_diame3,     nu4_load1_diame3,      Fba4_load1_diame3,
Fdyn4_load1_diame3,        Fbrake4_load1_diame3, Fra4_load1_diame3,
Fg4_load1_diame3,          Fext4_load1_diame3,   Fb4_load1_diame3,
Fttotal4_load1_diame3,     acc4_load1_diame3,    s4_load1_diame3,
time_eq4_load1_diame3,
distance_eq4_load1_diame3,  deceleration_eq4_load1_diame3,
velocity4_load2_diame1,     time4_load2_diame1,   pressure4_load2_diame1,
Fc4_load2_diame1,          Fb4_load2_diame1,     nu4_load2_diame1,      Fba4_load2_diame1,
Fdyn4_load2_diame1,        Fbrake4_load2_diame1, Fra4_load2_diame1,
Fg4_load2_diame1,          Fext4_load2_diame1,   Fb4_load2_diame1,
Fttotal4_load2_diame1,     acc4_load2_diame1,    s4_load2_diame1,
time_eq4_load2_diame1,
distance_eq4_load2_diame1,  deceleration_eq4_load2_diame1,
velocity4_load2_diame2,     time4_load2_diame2,   pressure4_load2_diame2,
Fc4_load2_diame2,          Fb4_load2_diame2,     nu4_load2_diame2,      Fba4_load2_diame2,
Fdyn4_load2_diame2,        Fbrake4_load2_diame2, Fra4_load2_diame2,
Fg4_load2_diame2,          Fext4_load2_diame2,   Fb4_load2_diame2,
Fttotal4_load2_diame2,     acc4_load2_diame2,    s4_load2_diame2,
time_eq4_load2_diame2,
distance_eq4_load2_diame2,  deceleration_eq4_load2_diame2,
velocity4_load2_diame3,     time4_load2_diame3,   pressure4_load2_diame3,
Fc4_load2_diame3,          Fb4_load2_diame3,     nu4_load2_diame3,      Fba4_load2_diame3,
Fdyn4_load2_diame3,        Fbrake4_load2_diame3, Fra4_load2_diame3,
Fg4_load2_diame3,          Fext4_load2_diame3,   Fb4_load2_diame3,
Fttotal4_load2_diame3,     acc4_load2_diame3,    s4_load2_diame3,
time_eq4_load2_diame3,
distance_eq4_load2_diame3,  deceleration_eq4_load2_diame3,
velocity4_load3_diame1,     time4_load3_diame1,   pressure4_load3_diame1,
Fc4_load3_diame1,          Fb4_load3_diame1,     nu4_load3_diame1,      Fba4_load3_diame1,
Fdyn4_load3_diame1,        Fbrake4_load3_diame1, Fra4_load3_diame1,
```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

Fg4_load3_diame1, Fext4_load3_diame1, Fb4_load3_diame1,
Ftotal4_load3_diame1, acc4_load3_diame1, s4_load3_diame1,
time_eq4_load3_diame1,
distance_eq4_load3_diame1, deceleration_eq4_load3_diame1,
velocity4_load3_diame2, time4_load3_diame2, pressure4_load3_diame2,
Fc4_load3_diame2, Fb4_load3_diame2, nu4_load3_diame2, Fba4_load3_diame2,
Fdyn4_load3_diame2, Fbrake4_load3_diame2, Fra4_load3_diame2,
Fg4_load3_diame2, Fext4_load3_diame2, Fb4_load3_diame2,
Ftotal4_load3_diame2, acc4_load3_diame2, s4_load3_diame2,
time_eq4_load3_diame2,
distance_eq4_load3_diame2, deceleration_eq4_load3_diame2,
velocity4_load3_diame3, time4_load3_diame3, pressure4_load3_diame3,
Fc4_load3_diame3, Fb4_load3_diame3, nu4_load3_diame3, Fba4_load3_diame3,
Fdyn4_load3_diame3, Fbrake4_load3_diame3, Fra4_load3_diame3,
Fg4_load3_diame3, Fext4_load3_diame3, Fb4_load3_diame3,
Ftotal4_load3_diame3, acc4_load3_diame3, s4_load3_diame3,
time_eq4_load3_diame3,
distance_eq4_load3_diame3, deceleration_eq4_load3_diame3,
velocity4_load4_diame1, time4_load4_diame1, pressure4_load4_diame1,
Fc4_load4_diame1, Fb4_load4_diame1, nu4_load4_diame1, Fba4_load4_diame1,
Fdyn4_load4_diame1, Fbrake4_load4_diame1, Fra4_load4_diame1,
Fg4_load4_diame1, Fext4_load4_diame1, Fb4_load4_diame1,
Ftotal4_load4_diame1, acc4_load4_diame1, s4_load4_diame1,
time_eq4_load4_diame1,
distance_eq4_load4_diame1, deceleration_eq4_load4_diame1,
velocity4_load4_diame2, time4_load4_diame2, pressure4_load4_diame2,
Fc4_load4_diame2, Fb4_load4_diame2, nu4_load4_diame2, Fba4_load4_diame2,
Fdyn4_load4_diame2, Fbrake4_load4_diame2, Fra4_load4_diame2,
Fg4_load4_diame2, Fext4_load4_diame2, Fb4_load4_diame2,
Ftotal4_load4_diame2, acc4_load4_diame2, s4_load4_diame2,
time_eq4_load4_diame2,
distance_eq4_load4_diame2, deceleration_eq4_load4_diame2,
velocity4_load4_diame3, time4_load4_diame3, pressure4_load4_diame3,
Fc4_load4_diame3, Fb4_load4_diame3, nu4_load4_diame3, Fba4_load4_diame3,
Fdyn4_load4_diame3, Fbrake4_load4_diame3, Fra4_load4_diame3,
Fg4_load4_diame3, Fext4_load4_diame3, Fb4_load4_diame3,
Ftotal4_load4_diame3, acc4_load4_diame3, s4_load4_diame3,
time_eq4_load4_diame3,
distance_eq4_load4_diame3, deceleration_eq4_load4_diame3, velocity4
_load5_diame1, time4_load5_diame1, pressure4_load5_diame1,
Fc4_load5_diame1, Fb4_load5_diame1, nu4_load5_diame1, Fba4_load5_diame1,
Fdyn4_load5_diame1, Fbrake4_load5_diame1, Fra4_load5_diame1,
Fg4_load5_diame1, Fext4_load5_diame1, Fb4_load5_diame1,
Ftotal4_load5_diame1, acc4_load5_diame1, s4_load5_diame1,
time_eq4_load5_diame1,
distance_eq4_load5_diame1, deceleration_eq4_load5_diame1,
velocity4_load5_diame2, time4_load5_diame2, pressure4_load5_diame2,
Fc4_load5_diame2, Fb4_load5_diame2, nu4_load5_diame2, Fba4_load5_diame2,
Fdyn4_load5_diame2, Fbrake4_load5_diame2, Fra4_load5_diame2,
Fg4_load5_diame2, Fext4_load5_diame2, Fb4_load5_diame2,
Ftotal4_load5_diame2, acc4_load5_diame2, s4_load5_diame2,
time_eq4_load5_diame2,
distance_eq4_load5_diame2, deceleration_eq4_load5_diame2,
velocity4_load5_diame3, time4_load5_diame3, pressure4_load5_diame3,
Fc4_load5_diame3, Fb4_load5_diame3, nu4_load5_diame3, Fba4_load5_diame3,
Fdyn4_load5_diame3, Fbrake4_load5_diame3, Fra4_load5_diame3,

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
Fg4_load5_diame3,          Fext4_load5_diame3,          Fb4_load5_diame3,
Ftotal4_load5_diame3,     acc4_load5_diame3,          s4_load5_diame3,
time_eq4_load5_diame3,
distance_eq4_load5_diame3, deceleration_eq4_load5_diame3, velocity4,
time4, pressure4, Fc4, Fb4, nu4, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake4, Fra4, Fg1, Fg2, Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext4, FB4,
Ftotal4, acc4, s4, time_eq4, distance_eq4, deceleration_eq4] =
load_diam(stp, x, condition, sav, GeneralitiesVehicleComp_Data,
NominalMass_Data,          DynamicMass_Data,          WheelDiameter_Data,
VehicleResistance_Data,    FrictionBrakeForces_Data,
DynamicBrakeForces_Data, TrackBrake_Data, Initial_Data, Degraded_Data_4,
Pressure_Vel_Data);
```

```
Fba1_cond4 = Fba1;
Fba2_cond4 = Fba2;
Fba3_cond4 = Fba3;
Fba4_cond4 = Fba4;
Fba5_cond4 = Fba5;
Fba6_cond4 = Fba6;
Fba7_cond4 = Fba7;
Fba8_cond4 = Fba8;
Fba9_cond4 = Fba9;
```

```
Fdyn1_cond4 = Fdyn1;
Fdyn2_cond4 = Fdyn2;
Fdyn3_cond4 = Fdyn3;
Fdyn4_cond4 = Fdyn4;
Fdyn5_cond4 = Fdyn5;
Fdyn6_cond4 = Fdyn6;
Fdyn7_cond4 = Fdyn7;
Fdyn8_cond4 = Fdyn8;
Fdyn9_cond4 = Fdyn9;
```

```
Fra1_cond4 = Fra1;
Fra2_cond4 = Fra2;
Fra3_cond4 = Fra3;
Fra4_cond4 = Fra4;
Fra5_cond4 = Fra5;
Fra6_cond4 = Fra6;
Fra7_cond4 = Fra7;
Fra8_cond4 = Fra8;
Fra9_cond4 = Fra9;
```

```
Fg1_cond4 = Fg1;
Fg2_cond4 = Fg2;
Fg3_cond4 = Fg3;
Fg4_cond4 = Fg4;
Fg5_cond4 = Fg5;
Fg6_cond4 = Fg6;
Fg7_cond4 = Fg7;
Fg8_cond4 = Fg8;
Fg9_cond4 = Fg9;
```

```
s_load1_diame1_1 = s4_load1_diame1;
s_load1_diame2_1 = s4_load1_diame2;
```

```

s_load1_diame3_1 = s4_load1_diame3;
s_load2_diame1_1 = s4_load2_diame1;
s_load2_diame2_1 = s4_load2_diame2;
s_load2_diame3_1 = s4_load2_diame3;
s_load3_diame1_1 = s4_load3_diame1;
s_load3_diame2_1 = s4_load3_diame2;
s_load3_diame3_1 = s4_load3_diame3;
s_load4_diame1_1 = s4_load4_diame1;
s_load4_diame2_1 = s4_load4_diame2;
s_load4_diame3_1 = s4_load4_diame3;
s_load5_diame1_1 = s4_load5_diame1;
s_load5_diame2_1 = s4_load5_diame2;
s_load5_diame3_1 = s4_load5_diame3;

if length(s_load1_diame1_2) ~= 0
    dev = abs((s_load1_diame1_2(end) -
s_load1_diame1_1(end))/s_load1_diame1_1(end));
    deviation4_load1_diame1(condition) = dev;
end
if length(s_load1_diame2_2) ~= 0
    dev = abs((s_load1_diame2_2(end) -
s_load1_diame2_1(end))/s_load1_diame2_1(end));
    deviation4_load1_diame2(condition) = dev;
end
if length(s_load1_diame3_2) ~= 0
    dev = abs((s_load1_diame3_2(end) -
s_load1_diame3_1(end))/s_load1_diame3_1(end));
    deviation4_load1_diame3(condition) = dev;
end
if length(s_load2_diame1_2) ~= 0
    dev = abs((s_load2_diame1_2(end) -
s_load2_diame1_1(end))/s_load2_diame1_1(end));
    deviation4_load2_diame1(condition) = dev;
end
if length(s_load2_diame2_2) ~= 0
    dev = abs((s_load2_diame2_2(end) -
s_load2_diame2_1(end))/s_load2_diame2_1(end));
    deviation4_load2_diame2(condition) = dev;
end
if length(s_load2_diame3_2) ~= 0
    dev = abs((s_load2_diame3_2(end) -
s_load2_diame3_1(end))/s_load2_diame3_1(end));
    deviation4_load2_diame3(condition) = dev;
end
if length(s_load3_diame1_2) ~= 0
    dev = abs((s_load3_diame1_2(end) -
s_load3_diame1_1(end))/s_load3_diame1_1(end));
    deviation4_load3_diame1(condition) = dev;
end
if length(s_load3_diame2_2) ~= 0
    dev = abs((s_load3_diame2_2(end) -
s_load3_diame2_1(end))/s_load3_diame2_1(end));
    deviation4_load3_diame2(condition) = dev;
end
if length(s_load3_diame3_2) ~= 0

```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un vehículo ferroviario

```

                                dev      =      abs((s_load3_diame3_2(end) -
s_load3_diame3_1(end))/s_load3_diame3_1(end));
                                deviation4_load3_diame3(condition) = dev;
                                end
                                if length(s_load4_diame1_2) ~= 0
                                        dev      =      abs((s_load4_diame1_2(end) -
s_load4_diame1_1(end))/s_load4_diame1_1(end));
                                        deviation4_load4_diame1(condition) = dev;
                                end
                                if length(s_load4_diame2_2) ~= 0
                                        dev      =      abs((s_load4_diame2_2(end) -
s_load4_diame2_1(end))/s_load4_diame2_1(end));
                                        deviation4_load4_diame2(condition) = dev;
                                end
                                if length(s_load4_diame3_2) ~= 0
                                        dev      =      abs((s_load4_diame3_2(end) -
s_load4_diame3_1(end))/s_load4_diame3_1(end));
                                        deviation4_load4_diame3(condition) = dev;
                                end
                                if length(s_load5_diame1_2) ~= 0
                                        dev      =      abs((s_load5_diame1_2(end) -
s_load5_diame1_1(end))/s_load5_diame1_1(end));
                                        deviation4_load5_diame1(condition) = dev;
                                end
                                if length(s_load5_diame2_2) ~= 0
                                        dev      =      abs((s_load5_diame2_2(end) -
s_load5_diame2_1(end))/s_load5_diame2_1(end));
                                        deviation4_load5_diame2(condition) = dev;
                                end
                                if length(s_load5_diame3_2) ~= 0
                                        dev      =      abs((s_load5_diame3_2(end) -
s_load5_diame3_1(end))/s_load5_diame3_1(end));
                                        deviation4_load5_diame3(condition) = dev;
                                end
                                end
                                if condition == 5
                                        sav = 0;
                                [velocity5_load1_diame1,      time5_load1_diame1,
pressure5_load1_diame1,      Fc5_load1_diame1,      Fb5_load1_diame1,
nu5_load1_diame1,      Fba5_load1_diame1,      Fdyn5_load1_diame1,
Fbrake5_load1_diame1,      Fra5_load1_diame1,      Fg5_load1_diame1,
Fext5_load1_diame1,      Fb5_load1_diame1,      Ftotal5_load1_diame1,
acc5_load1_diame1,      s5_load1_diame1,      time_eq5_load1_diame1,
distance_eq5_load1_diame1,      deceleration_eq5_load1_diame1,
velocity5_load1_diame2,      time5_load1_diame2,      pressure5_load1_diame2,
Fc5_load1_diame2,      Fb5_load1_diame2,      nu5_load1_diame2,      Fba5_load1_diame2,
Fdyn5_load1_diame2,      Fbrake5_load1_diame2,      Fra5_load1_diame2,
Fg5_load1_diame2,      Fext5_load1_diame2,      Fb5_load1_diame2,
Ftotal5_load1_diame2,      acc5_load1_diame2,      s5_load1_diame2,
time_eq5_load1_diame2,
distance_eq5_load1_diame2,      deceleration_eq5_load1_diame2,
velocity5_load1_diame3,      time5_load1_diame3,      pressure5_load1_diame3,
Fc5_load1_diame3,      Fb5_load1_diame3,      nu5_load1_diame3,      Fba5_load1_diame3,
Fdyn5_load1_diame3,      Fbrake5_load1_diame3,      Fra5_load1_diame3,
Fg5_load1_diame3,      Fext5_load1_diame3,      Fb5_load1_diame3,
Ftotal5_load1_diame3,      acc5_load1_diame3,      s5_load1_diame3,

```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
time_eq5_load1_diame3,
distance_eq5_load1_diame3,                deceleration_eq5_load1_diame3,
velocity5_load2_diame1,  time5_load2_diame1,  pressure5_load2_diame1,
Fc5_load2_diame1,  Fb5_load2_diame1,  nu5_load2_diame1,  Fba5_load2_diame1,
Fdyn5_load2_diame1,  Fbrake5_load2_diame1,  Fra5_load2_diame1,
Fg5_load2_diame1,  Fext5_load2_diame1,  Fb5_load2_diame1,
Ftotal5_load2_diame1,  acc5_load2_diame1,  s5_load2_diame1,
time_eq5_load2_diame1,
distance_eq5_load2_diame1,                deceleration_eq5_load2_diame1,
velocity5_load2_diame2,  time5_load2_diame2,  pressure5_load2_diame2,
Fc5_load2_diame2,  Fb5_load2_diame2,  nu5_load2_diame2,  Fba5_load2_diame2,
Fdyn5_load2_diame2,  Fbrake5_load2_diame2,  Fra5_load2_diame2,
Fg5_load2_diame2,  Fext5_load2_diame2,  Fb5_load2_diame2,
Ftotal5_load2_diame2,  acc5_load2_diame2,  s5_load2_diame2,
time_eq5_load2_diame2,
distance_eq5_load2_diame2,                deceleration_eq5_load2_diame2,
velocity5_load2_diame3,  time5_load2_diame3,  pressure5_load2_diame3,
Fc5_load2_diame3,  Fb5_load2_diame3,  nu5_load2_diame3,  Fba5_load2_diame3,
Fdyn5_load2_diame3,  Fbrake5_load2_diame3,  Fra5_load2_diame3,
Fg5_load2_diame3,  Fext5_load2_diame3,  Fb5_load2_diame3,
Ftotal5_load2_diame3,  acc5_load2_diame3,  s5_load2_diame3,
time_eq5_load2_diame3,
distance_eq5_load2_diame3,                deceleration_eq5_load2_diame3,
velocity5_load3_diame1,  time5_load3_diame1,  pressure5_load3_diame1,
Fc5_load3_diame1,  Fb5_load3_diame1,  nu5_load3_diame1,  Fba5_load3_diame1,
Fdyn5_load3_diame1,  Fbrake5_load3_diame1,  Fra5_load3_diame1,
Fg5_load3_diame1,  Fext5_load3_diame1,  Fb5_load3_diame1,
Ftotal5_load3_diame1,  acc5_load3_diame1,  s5_load3_diame1,
time_eq5_load3_diame1,
distance_eq5_load3_diame1,                deceleration_eq5_load3_diame1,
velocity5_load3_diame2,  time5_load3_diame2,  pressure5_load3_diame2,
Fc5_load3_diame2,  Fb5_load3_diame2,  nu5_load3_diame2,  Fba5_load3_diame2,
Fdyn5_load3_diame2,  Fbrake5_load3_diame2,  Fra5_load3_diame2,
Fg5_load3_diame2,  Fext5_load3_diame2,  Fb5_load3_diame2,
Ftotal5_load3_diame2,  acc5_load3_diame2,  s5_load3_diame2,
time_eq5_load3_diame2,
distance_eq5_load3_diame2,                deceleration_eq5_load3_diame2,
velocity5_load3_diame3,  time5_load3_diame3,  pressure5_load3_diame3,
Fc5_load3_diame3,  Fb5_load3_diame3,  nu5_load3_diame3,  Fba5_load3_diame3,
Fdyn5_load3_diame3,  Fbrake5_load3_diame3,  Fra5_load3_diame3,
Fg5_load3_diame3,  Fext5_load3_diame3,  Fb5_load3_diame3,
Ftotal5_load3_diame3,  acc5_load3_diame3,  s5_load3_diame3,
time_eq5_load3_diame3,
distance_eq5_load3_diame3,                deceleration_eq5_load3_diame3,
velocity5_load4_diame1,  time5_load4_diame1,  pressure5_load4_diame1,
Fc5_load4_diame1,  Fb5_load4_diame1,  nu5_load4_diame1,  Fba5_load4_diame1,
Fdyn5_load4_diame1,  Fbrake5_load4_diame1,  Fra5_load4_diame1,
Fg5_load4_diame1,  Fext5_load4_diame1,  Fb5_load4_diame1,
Ftotal5_load4_diame1,  acc5_load4_diame1,  s5_load4_diame1,
time_eq5_load4_diame1,
distance_eq5_load4_diame1,                deceleration_eq5_load4_diame1,
velocity5_load4_diame2,  time5_load4_diame2,  pressure5_load4_diame2,
Fc5_load4_diame2,  Fb5_load4_diame2,  nu5_load4_diame2,  Fba5_load4_diame2,
Fdyn5_load4_diame2,  Fbrake5_load4_diame2,  Fra5_load4_diame2,
Fg5_load4_diame2,  Fext5_load4_diame2,  Fb5_load4_diame2,
Ftotal5_load4_diame2,  acc5_load4_diame2,  s5_load4_diame2,
```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```
time_eq5_load4_diame2,
distance_eq5_load4_diame2,          deceleration_eq5_load4_diame2,
velocity5_load4_diame3,  time5_load4_diame3,  pressure5_load4_diame3,
Fc5_load4_diame3,  Fb5_load4_diame3,  nu5_load4_diame3,  Fba5_load4_diame3,
Fdyn5_load4_diame3,  Fbrake5_load4_diame3,  Fra5_load4_diame3,
Fg5_load4_diame3,  Fext5_load4_diame3,  Fb5_load4_diame3,
Ftotal5_load4_diame3,  acc5_load4_diame3,  s5_load4_diame3,
time_eq5_load4_diame3,
distance_eq5_load4_diame3,  deceleration_eq5_load4_diame3,  velocity5
_load5_diame1,  time5_load5_diame1,  pressure5_load5_diame1,
Fc5_load5_diame1,  Fb5_load5_diame1,  nu5_load5_diame1,  Fba5_load5_diame1,
Fdyn5_load5_diame1,  Fbrake5_load5_diame1,  Fra5_load5_diame1,
Fg5_load5_diame1,  Fext5_load5_diame1,  Fb5_load5_diame1,
Ftotal5_load5_diame1,  acc5_load5_diame1,  s5_load5_diame1,
time_eq5_load5_diame1,
distance_eq5_load5_diame1,          deceleration_eq5_load5_diame1,
velocity5_load5_diame2,  time5_load5_diame2,  pressure5_load5_diame2,
Fc5_load5_diame2,  Fb5_load5_diame2,  nu5_load5_diame2,  Fba5_load5_diame2,
Fdyn5_load5_diame2,  Fbrake5_load5_diame2,  Fra5_load5_diame2,
Fg5_load5_diame2,  Fext5_load5_diame2,  Fb5_load5_diame2,
Ftotal5_load5_diame2,  acc5_load5_diame2,  s5_load5_diame2,
time_eq5_load5_diame2,
distance_eq5_load5_diame2,          deceleration_eq5_load5_diame2,
velocity5_load5_diame3,  time5_load5_diame3,  pressure5_load5_diame3,
Fc5_load5_diame3,  Fb5_load5_diame3,  nu5_load5_diame3,  Fba5_load5_diame3,
Fdyn5_load5_diame3,  Fbrake5_load5_diame3,  Fra5_load5_diame3,
Fg5_load5_diame3,  Fext5_load5_diame3,  Fb5_load5_diame3,
Ftotal5_load5_diame3,  acc5_load5_diame3,  s5_load5_diame3,
time_eq5_load5_diame3,
distance_eq5_load5_diame3,  deceleration_eq5_load5_diame3,  velocity5,
time5, pressure5, Fc5, Fb5, nu5, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake5, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9, Fg1, Fg2,
Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext5, FB5, Ftotal5, acc5, s5,
time_eq5,  distance_eq5,  deceleration_eq5] = load_diam(stp2,  x,
condition,  sav,  GeneralitiesVehicleComp_Data,  NominalMass_Data,
DynamicMass_Data,  WheelDiameter_Data,  VehicleResistance_Data,
FrictionBrakeForces_Data,  DynamicBrakeForces_Data,  TrackBrake_Data,
Initial_Data, Degraded_Data_5, Pressure_Vel_Data);

s_load1_diame1_2 = s5_load1_diame1;
s_load1_diame2_2 = s5_load1_diame2;
s_load1_diame3_2 = s5_load1_diame3;
s_load2_diame1_2 = s5_load2_diame1;
s_load2_diame2_2 = s5_load2_diame2;
s_load2_diame3_2 = s5_load2_diame3;
s_load3_diame1_2 = s5_load3_diame1;
s_load3_diame2_2 = s5_load3_diame2;
s_load3_diame3_2 = s5_load3_diame3;
s_load4_diame1_2 = s5_load4_diame1;
s_load4_diame2_2 = s5_load4_diame2;
s_load4_diame3_2 = s5_load4_diame3;
s_load5_diame1_2 = s5_load5_diame1;
s_load5_diame2_2 = s5_load5_diame2;
s_load5_diame3_2 = s5_load5_diame3;
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
sav = 1;
[velocity5_load1_diame1,      time5_load1_diame1,
pressure5_load1_diame1,      Fc5_load1_diame1,      Fb5_load1_diame1,
nu5_load1_diame1,           Fba5_load1_diame1,      Fdyn5_load1_diame1,
Fbrake5_load1_diame1,       Fra5_load1_diame1,      Fg5_load1_diame1,
Fext5_load1_diame1,         Fb5_load1_diame1,      Ftotal5_load1_diame1,
acc5_load1_diame1,          s5_load1_diame1,        time_eq5_load1_diame1,
distance_eq5_load1_diame1,   deceleration_eq5_load1_diame1,
velocity5_load1_diame2,      time5_load1_diame2,      pressure5_load1_diame2,
Fc5_load1_diame2, Fb5_load1_diame2, nu5_load1_diame2, Fba5_load1_diame2,
Fdyn5_load1_diame2, Fbrake5_load1_diame2, Fra5_load1_diame2,
Fg5_load1_diame2, Fext5_load1_diame2, Fb5_load1_diame2,
Ftotal5_load1_diame2, acc5_load1_diame2, s5_load1_diame2,
time_eq5_load1_diame2,
distance_eq5_load1_diame2,   deceleration_eq5_load1_diame2,
velocity5_load1_diame3,      time5_load1_diame3,      pressure5_load1_diame3,
Fc5_load1_diame3, Fb5_load1_diame3, nu5_load1_diame3, Fba5_load1_diame3,
Fdyn5_load1_diame3, Fbrake5_load1_diame3, Fra5_load1_diame3,
Fg5_load1_diame3, Fext5_load1_diame3, Fb5_load1_diame3,
Ftotal5_load1_diame3, acc5_load1_diame3, s5_load1_diame3,
time_eq5_load1_diame3,
distance_eq5_load1_diame3,   deceleration_eq5_load1_diame3,
velocity5_load2_diame1,      time5_load2_diame1,      pressure5_load2_diame1,
Fc5_load2_diame1, Fb5_load2_diame1, nu5_load2_diame1, Fba5_load2_diame1,
Fdyn5_load2_diame1, Fbrake5_load2_diame1, Fra5_load2_diame1,
Fg5_load2_diame1, Fext5_load2_diame1, Fb5_load2_diame1,
Ftotal5_load2_diame1, acc5_load2_diame1, s5_load2_diame1,
time_eq5_load2_diame1,
distance_eq5_load2_diame1,   deceleration_eq5_load2_diame1,
velocity5_load2_diame2,      time5_load2_diame2,      pressure5_load2_diame2,
Fc5_load2_diame2, Fb5_load2_diame2, nu5_load2_diame2, Fba5_load2_diame2,
Fdyn5_load2_diame2, Fbrake5_load2_diame2, Fra5_load2_diame2,
Fg5_load2_diame2, Fext5_load2_diame2, Fb5_load2_diame2,
Ftotal5_load2_diame2, acc5_load2_diame2, s5_load2_diame2,
time_eq5_load2_diame2,
distance_eq5_load2_diame2,   deceleration_eq5_load2_diame2,
velocity5_load2_diame3,      time5_load2_diame3,      pressure5_load2_diame3,
Fc5_load2_diame3, Fb5_load2_diame3, nu5_load2_diame3, Fba5_load2_diame3,
Fdyn5_load2_diame3, Fbrake5_load2_diame3, Fra5_load2_diame3,
Fg5_load2_diame3, Fext5_load2_diame3, Fb5_load2_diame3,
Ftotal5_load2_diame3, acc5_load2_diame3, s5_load2_diame3,
time_eq5_load2_diame3,
distance_eq5_load2_diame3,   deceleration_eq5_load2_diame3,
velocity5_load3_diame1,      time5_load3_diame1,      pressure5_load3_diame1,
Fc5_load3_diame1, Fb5_load3_diame1, nu5_load3_diame1, Fba5_load3_diame1,
Fdyn5_load3_diame1, Fbrake5_load3_diame1, Fra5_load3_diame1,
Fg5_load3_diame1, Fext5_load3_diame1, Fb5_load3_diame1,
Ftotal5_load3_diame1, acc5_load3_diame1, s5_load3_diame1,
time_eq5_load3_diame1,
distance_eq5_load3_diame1,   deceleration_eq5_load3_diame1,
velocity5_load3_diame2,      time5_load3_diame2,      pressure5_load3_diame2,
Fc5_load3_diame2, Fb5_load3_diame2, nu5_load3_diame2, Fba5_load3_diame2,
Fdyn5_load3_diame2, Fbrake5_load3_diame2, Fra5_load3_diame2,
Fg5_load3_diame2, Fext5_load3_diame2, Fb5_load3_diame2,
Ftotal5_load3_diame2, acc5_load3_diame2, s5_load3_diame2,
time_eq5_load3_diame2,
```


**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```
distance_eq5_load3_diame2,          deceleration_eq5_load3_diame2,
velocity5_load3_diame3,    time5_load3_diame3,    pressure5_load3_diame3,
Fc5_load3_diame3,    Fb5_load3_diame3,    nu5_load3_diame3,    Fba5_load3_diame3,
Fdyn5_load3_diame3,    Fbrake5_load3_diame3,    Fra5_load3_diame3,
Fg5_load3_diame3,    Fext5_load3_diame3,    Fb5_load3_diame3,
Ftotal5_load3_diame3,    acc5_load3_diame3,    s5_load3_diame3,
time_eq5_load3_diame3,
distance_eq5_load3_diame3,          deceleration_eq5_load3_diame3,
velocity5_load4_diame1,    time5_load4_diame1,    pressure5_load4_diame1,
Fc5_load4_diame1,    Fb5_load4_diame1,    nu5_load4_diame1,    Fba5_load4_diame1,
Fdyn5_load4_diame1,    Fbrake5_load4_diame1,    Fra5_load4_diame1,
Fg5_load4_diame1,    Fext5_load4_diame1,    Fb5_load4_diame1,
Ftotal5_load4_diame1,    acc5_load4_diame1,    s5_load4_diame1,
time_eq5_load4_diame1,
distance_eq5_load4_diame1,          deceleration_eq5_load4_diame1,
velocity5_load4_diame2,    time5_load4_diame2,    pressure5_load4_diame2,
Fc5_load4_diame2,    Fb5_load4_diame2,    nu5_load4_diame2,    Fba5_load4_diame2,
Fdyn5_load4_diame2,    Fbrake5_load4_diame2,    Fra5_load4_diame2,
Fg5_load4_diame2,    Fext5_load4_diame2,    Fb5_load4_diame2,
Ftotal5_load4_diame2,    acc5_load4_diame2,    s5_load4_diame2,
time_eq5_load4_diame2,
distance_eq5_load4_diame2,          deceleration_eq5_load4_diame2,
velocity5_load4_diame3,    time5_load4_diame3,    pressure5_load4_diame3,
Fc5_load4_diame3,    Fb5_load4_diame3,    nu5_load4_diame3,    Fba5_load4_diame3,
Fdyn5_load4_diame3,    Fbrake5_load4_diame3,    Fra5_load4_diame3,
Fg5_load4_diame3,    Fext5_load4_diame3,    Fb5_load4_diame3,
Ftotal5_load4_diame3,    acc5_load4_diame3,    s5_load4_diame3,
time_eq5_load4_diame3,
distance_eq5_load4_diame3,    deceleration_eq5_load4_diame3,    velocity5
_load5_diame1,    time5_load5_diame1,    pressure5_load5_diame1,
Fc5_load5_diame1,    Fb5_load5_diame1,    nu5_load5_diame1,    Fba5_load5_diame1,
Fdyn5_load5_diame1,    Fbrake5_load5_diame1,    Fra5_load5_diame1,
Fg5_load5_diame1,    Fext5_load5_diame1,    Fb5_load5_diame1,
Ftotal5_load5_diame1,    acc5_load5_diame1,    s5_load5_diame1,
time_eq5_load5_diame1,
distance_eq5_load5_diame1,          deceleration_eq5_load5_diame1,
velocity5_load5_diame2,    time5_load5_diame2,    pressure5_load5_diame2,
Fc5_load5_diame2,    Fb5_load5_diame2,    nu5_load5_diame2,    Fba5_load5_diame2,
Fdyn5_load5_diame2,    Fbrake5_load5_diame2,    Fra5_load5_diame2,
Fg5_load5_diame2,    Fext5_load5_diame2,    Fb5_load5_diame2,
Ftotal5_load5_diame2,    acc5_load5_diame2,    s5_load5_diame2,
time_eq5_load5_diame2,
distance_eq5_load5_diame2,          deceleration_eq5_load5_diame2,
velocity5_load5_diame3,    time5_load5_diame3,    pressure5_load5_diame3,
Fc5_load5_diame3,    Fb5_load5_diame3,    nu5_load5_diame3,    Fba5_load5_diame3,
Fdyn5_load5_diame3,    Fbrake5_load5_diame3,    Fra5_load5_diame3,
Fg5_load5_diame3,    Fext5_load5_diame3,    Fb5_load5_diame3,
Ftotal5_load5_diame3,    acc5_load5_diame3,    s5_load5_diame3,
time_eq5_load5_diame3,
distance_eq5_load5_diame3,    deceleration_eq5_load5_diame3,    velocity5,
time5, pressure5, Fc5, Fb5, nu5, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake5, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9, Fg1, Fg2,
Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext5, FB5, Ftotal5, acc5, s5,
time_eq5, distance_eq5, deceleration_eq5] = load_diam(stp, x, condition,
sav, GeneralitiesVehicleComp_Data, NominalMass_Data, DynamicMass_Data,
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```
WheelDiameter_Data, VehicleResistance_Data, FrictionBrakeForces_Data,  
DynamicBrakeForces_Data, TrackBrake_Data, Initial_Data, Degraded_Data_5,  
Pressure_Vel_Data);
```

```
Fba1_cond5 = Fba1;  
Fba2_cond5 = Fba2;  
Fba3_cond5 = Fba3;  
Fba4_cond5 = Fba4;  
Fba5_cond5 = Fba5;  
Fba6_cond5 = Fba6;  
Fba7_cond5 = Fba7;  
Fba8_cond5 = Fba8;  
Fba9_cond5 = Fba9;
```

```
Fdyn1_cond5 = Fdyn1;  
Fdyn2_cond5 = Fdyn2;  
Fdyn3_cond5 = Fdyn3;  
Fdyn4_cond5 = Fdyn4;  
Fdyn5_cond5 = Fdyn5;  
Fdyn6_cond5 = Fdyn6;  
Fdyn7_cond5 = Fdyn7;  
Fdyn8_cond5 = Fdyn8;  
Fdyn9_cond5 = Fdyn9;
```

```
Fra1_cond5 = Fra1;  
Fra2_cond5 = Fra2;  
Fra3_cond5 = Fra3;  
Fra4_cond5 = Fra4;  
Fra5_cond5 = Fra5;  
Fra6_cond5 = Fra6;  
Fra7_cond5 = Fra7;  
Fra8_cond5 = Fra8;  
Fra9_cond5 = Fra9;
```

```
Fg1_cond5 = Fg1;  
Fg2_cond5 = Fg2;  
Fg3_cond5 = Fg3;  
Fg4_cond5 = Fg4;  
Fg5_cond5 = Fg5;  
Fg6_cond5 = Fg6;  
Fg7_cond5 = Fg7;  
Fg8_cond5 = Fg8;  
Fg9_cond5 = Fg9;
```

```
s_load1_diame1_1 = s5_load1_diame1;  
s_load1_diame2_1 = s5_load1_diame2;  
s_load1_diame3_1 = s5_load1_diame3;  
s_load2_diame1_1 = s5_load2_diame1;  
s_load2_diame2_1 = s5_load2_diame2;  
s_load2_diame3_1 = s5_load2_diame3;  
s_load3_diame1_1 = s5_load3_diame1;  
s_load3_diame2_1 = s5_load3_diame2;  
s_load3_diame3_1 = s5_load3_diame3;  
s_load4_diame1_1 = s5_load4_diame1;  
s_load4_diame2_1 = s5_load4_diame2;  
s_load4_diame3_1 = s5_load4_diame3;
```

```
s_load5_diame1_1 = s5_load5_diame1;
s_load5_diame2_1 = s5_load5_diame2;
s_load5_diame3_1 = s5_load5_diame3;

if length(s_load1_diame1_2) ~= 0
    dev = abs((s_load1_diame1_2(end) -
s_load1_diame1_1(end))/s_load1_diame1_1(end));
    deviation5_load1_diame1(condition) = dev;
end
if length(s_load1_diame2_2) ~= 0
    dev = abs((s_load1_diame2_2(end) -
s_load1_diame2_1(end))/s_load1_diame2_1(end));
    deviation5_load1_diame2(condition) = dev;
end
if length(s_load1_diame3_2) ~= 0
    dev = abs((s_load1_diame3_2(end) -
s_load1_diame3_1(end))/s_load1_diame3_1(end));
    deviation5_load1_diame3(condition) = dev;
end
if length(s_load2_diame1_2) ~= 0
    dev = abs((s_load2_diame1_2(end) -
s_load2_diame1_1(end))/s_load2_diame1_1(end));
    deviation5_load2_diame1(condition) = dev;
end
if length(s_load2_diame2_2) ~= 0
    dev = abs((s_load2_diame2_2(end) -
s_load2_diame2_1(end))/s_load2_diame2_1(end));
    deviation5_load2_diame2(condition) = dev;
end
if length(s_load2_diame3_2) ~= 0
    dev = abs((s_load2_diame3_2(end) -
s_load2_diame3_1(end))/s_load2_diame3_1(end));
    deviation5_load2_diame3(condition) = dev;
end
if length(s_load3_diame1_2) ~= 0
    dev = abs((s_load3_diame1_2(end) -
s_load3_diame1_1(end))/s_load3_diame1_1(end));
    deviation5_load3_diame1(condition) = dev;
end
if length(s_load3_diame2_2) ~= 0
    dev = abs((s_load3_diame2_2(end) -
s_load3_diame2_1(end))/s_load3_diame2_1(end));
    deviation5_load3_diame2(condition) = dev;
end
if length(s_load3_diame3_2) ~= 0
    dev = abs((s_load3_diame3_2(end) -
s_load3_diame3_1(end))/s_load3_diame3_1(end));
    deviation5_load3_diame3(condition) = dev;
end
if length(s_load4_diame1_2) ~= 0
    dev = abs((s_load4_diame1_2(end) -
s_load4_diame1_1(end))/s_load4_diame1_1(end));
    deviation5_load4_diame1(condition) = dev;
end
if length(s_load4_diame2_2) ~= 0
```

```

                                dev      =      abs((s_load4_diame2_2(end) -
s_load4_diame2_1(end))/s_load4_diame2_1(end));
                                deviation5_load4_diame2(condition) = dev;
                                end
                                if length(s_load4_diame3_2) ~= 0
                                        dev      =      abs((s_load4_diame3_2(end) -
s_load4_diame3_1(end))/s_load4_diame3_1(end));
                                        deviation5_load4_diame3(condition) = dev;
                                end
                                if length(s_load5_diame1_2) ~= 0
                                        dev      =      abs((s_load5_diame1_2(end) -
s_load5_diame1_1(end))/s_load5_diame1_1(end));
                                        deviation5_load5_diame1(condition) = dev;
                                end
                                if length(s_load5_diame2_2) ~= 0
                                        dev      =      abs((s_load5_diame2_2(end) -
s_load5_diame2_1(end))/s_load5_diame2_1(end));
                                        deviation5_load5_diame2(condition) = dev;
                                end
                                if length(s_load5_diame3_2) ~= 0
                                        dev      =      abs((s_load5_diame3_2(end) -
s_load5_diame3_1(end))/s_load5_diame3_1(end));
                                        deviation5_load5_diame3(condition) = dev;
                                end
                                end
                                end
end
end

```

SCRIPT 3. LOAD_DIAM

```

function [velocity1_load1_diame1,          time1_load1_diame1,
pressure1_load1_diame1,          Fc1_load1_diame1,          Fb1_load1_diame1,
nu1_load1_diame1,          Fba1_load1_diame1,          Fdyn1_load1_diame1,
Fbrake1_load1_diame1,          Fra1_load1_diame1,          Fg1_load1_diame1,
Fext1_load1_diame1,          FB1_load1_diame1,          Ftotal1_load1_diame1,
acc1_load1_diame1,          s1_load1_diame1,          time_eq1_load1_diame1,
distance_eq1_load1_diame1,          deceleration_eq1_load1_diame1,
velocity1_load1_diame2,          time1_load1_diame2,          pressure1_load1_diame2,
Fc1_load1_diame2, Fb1_load1_diame2, nu1_load1_diame2, Fba1_load1_diame2,
Fdyn1_load1_diame2,          Fbrake1_load1_diame2,          Fra1_load1_diame2,
Fg1_load1_diame2,          Fext1_load1_diame2,          FB1_load1_diame2,
Ftotal1_load1_diame2,          acc1_load1_diame2,          s1_load1_diame2,
time_eq1_load1_diame2,
distance_eq1_load1_diame2,          deceleration_eq1_load1_diame2,
velocity1_load1_diame3,          time1_load1_diame3,          pressure1_load1_diame3,
Fc1_load1_diame3, Fb1_load1_diame3, nu1_load1_diame3, Fba1_load1_diame3,
Fdyn1_load1_diame3,          Fbrake1_load1_diame3,          Fra1_load1_diame3,
Fg1_load1_diame3,          Fext1_load1_diame3,          FB1_load1_diame3,
Ftotal1_load1_diame3,          acc1_load1_diame3,          s1_load1_diame3,

```

**Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario**

```
time_eq1_load1_diame3,
distance_eq1_load1_diame3,                deceleration_eq1_load1_diame3,
velocity1_load2_diame1,    time1_load2_diame1,    pressure1_load2_diame1,
Fc1_load2_diame1, Fb1_load2_diame1, nu1_load2_diame1, Fba1_load2_diame1,
Fdyn1_load2_diame1,    Fbrake1_load2_diame1,    Fra1_load2_diame1,
Fg1_load2_diame1,    Fext1_load2_diame1,    FB1_load2_diame1,
Ftotal1_load2_diame1,    acc1_load2_diame1,    s1_load2_diame1,
time_eq1_load2_diame1,
distance_eq1_load2_diame1,                deceleration_eq1_load2_diame1,
velocity1_load2_diame2,    time1_load2_diame2,    pressure1_load2_diame2,
Fc1_load2_diame2, Fb1_load2_diame2, nu1_load2_diame2, Fba1_load2_diame2,
Fdyn1_load2_diame2,    Fbrake1_load2_diame2,    Fra1_load2_diame2,
Fg1_load2_diame2,    Fext1_load2_diame2,    FB1_load2_diame2,
Ftotal1_load2_diame2,    acc1_load2_diame2,    s1_load2_diame2,
time_eq1_load2_diame2,
distance_eq1_load2_diame2,                deceleration_eq1_load2_diame2,
velocity1_load2_diame3,    time1_load2_diame3,    pressure1_load2_diame3,
Fc1_load2_diame3, Fb1_load2_diame3, nu1_load2_diame3, Fba1_load2_diame3,
Fdyn1_load2_diame3,    Fbrake1_load2_diame3,    Fra1_load2_diame3,
Fg1_load2_diame3,    Fext1_load2_diame3,    FB1_load2_diame3,
Ftotal1_load2_diame3,    acc1_load2_diame3,    s1_load2_diame3,
time_eq1_load2_diame3,
distance_eq1_load2_diame3,                deceleration_eq1_load2_diame3,
velocity1_load3_diame1,    time1_load3_diame1,    pressure1_load3_diame1,
Fc1_load3_diame1, Fb1_load3_diame1, nu1_load3_diame1, Fba1_load3_diame1,
Fdyn1_load3_diame1,    Fbrake1_load3_diame1,    Fra1_load3_diame1,
Fg1_load3_diame1,    Fext1_load3_diame1,    FB1_load3_diame1,
Ftotal1_load3_diame1,    acc1_load3_diame1,    s1_load3_diame1,
time_eq1_load3_diame1,
distance_eq1_load3_diame1,                deceleration_eq1_load3_diame1,
velocity1_load3_diame2,    time1_load3_diame2,    pressure1_load3_diame2,
Fc1_load3_diame2, Fb1_load3_diame2, nu1_load3_diame2, Fba1_load3_diame2,
Fdyn1_load3_diame2,    Fbrake1_load3_diame2,    Fra1_load3_diame2,
Fg1_load3_diame2,    Fext1_load3_diame2,    FB1_load3_diame2,
Ftotal1_load3_diame2,    acc1_load3_diame2,    s1_load3_diame2,
time_eq1_load3_diame2,
distance_eq1_load3_diame2,                deceleration_eq1_load3_diame2,
velocity1_load3_diame3,    time1_load3_diame3,    pressure1_load3_diame3,
Fc1_load3_diame3, Fb1_load3_diame3, nu1_load3_diame3, Fba1_load3_diame3,
Fdyn1_load3_diame3,    Fbrake1_load3_diame3,    Fra1_load3_diame3,
Fg1_load3_diame3,    Fext1_load3_diame3,    FB1_load3_diame3,
Ftotal1_load3_diame3,    acc1_load3_diame3,    s1_load3_diame3,
time_eq1_load3_diame3,
distance_eq1_load3_diame3,                deceleration_eq1_load3_diame3,
velocity1_load4_diame1,    time1_load4_diame1,    pressure1_load4_diame1,
Fc1_load4_diame1, Fb1_load4_diame1, nu1_load4_diame1, Fba1_load4_diame1,
Fdyn1_load4_diame1,    Fbrake1_load4_diame1,    Fra1_load4_diame1,
Fg1_load4_diame1,    Fext1_load4_diame1,    FB1_load4_diame1,
Ftotal1_load4_diame1,    acc1_load4_diame1,    s1_load4_diame1,
time_eq1_load4_diame1,
distance_eq1_load4_diame1,                deceleration_eq1_load4_diame1,
velocity1_load4_diame2,    time1_load4_diame2,    pressure1_load4_diame2,
Fc1_load4_diame2, Fb1_load4_diame2, nu1_load4_diame2, Fba1_load4_diame2,
Fdyn1_load4_diame2,    Fbrake1_load4_diame2,    Fra1_load4_diame2,
Fg1_load4_diame2,    Fext1_load4_diame2,    FB1_load4_diame2,
Ftotal1_load4_diame2,    acc1_load4_diame2,    s1_load4_diame2,
```

Diseño de una herramienta de cálculo para evaluar las prestaciones del sistema de frenado de un
vehículo ferroviario

```

time_eq1_load4_diame2,
distance_eq1_load4_diame2,          deceleration_eq1_load4_diame2,
velocity1_load4_diame3,    time1_load4_diame3,    pressure1_load4_diame3,
Fc1_load4_diame3, Fb1_load4_diame3, nul_load4_diame3, Fba1_load4_diame3,
Fdyn1_load4_diame3,    Fbrake1_load4_diame3,    Fra1_load4_diame3,
Fg1_load4_diame3,    Fext1_load4_diame3,    FB1_load4_diame3,
Ftotal1_load4_diame3,    acc1_load4_diame3,    s1_load4_diame3,
time_eq1_load4_diame3,
distance_eq1_load4_diame3,    deceleration_eq1_load4_diame3,    velocity1
_load5_diame1,    time1_load5_diame1,    pressure1_load5_diame1,
Fc1_load5_diame1, Fb1_load5_diame1, nul_load5_diame1, Fba1_load5_diame1,
Fdyn1_load5_diame1,    Fbrake1_load5_diame1,    Fra1_load5_diame1,
Fg1_load5_diame1,    Fext1_load5_diame1,    FB1_load5_diame1,
Ftotal1_load5_diame1,    acc1_load5_diame1,    s1_load5_diame1,
time_eq1_load5_diame1,
distance_eq1_load5_diame1,          deceleration_eq1_load5_diame1,
velocity1_load5_diame2,    time1_load5_diame2,    pressure1_load5_diame2,
Fc1_load5_diame2, Fb1_load5_diame2, nul_load5_diame2, Fba1_load5_diame2,
Fdyn1_load5_diame2,    Fbrake1_load5_diame2,    Fra1_load5_diame2,
Fg1_load5_diame2,    Fext1_load5_diame2,    FB1_load5_diame2,
Ftotal1_load5_diame2,    acc1_load5_diame2,    s1_load5_diame2,
time_eq1_load5_diame2,
distance_eq1_load5_diame2,          deceleration_eq1_load5_diame2,
velocity1_load5_diame3,    time1_load5_diame3,    pressure1_load5_diame3,
Fc1_load5_diame3, Fb1_load5_diame3, nul_load5_diame3, Fba1_load5_diame3,
Fdyn1_load5_diame3,    Fbrake1_load5_diame3,    Fra1_load5_diame3,
Fg1_load5_diame3,    Fext1_load5_diame3,    FB1_load5_diame3,
Ftotal1_load5_diame3,    acc1_load5_diame3,    s1_load5_diame3,
time_eq1_load5_diame3,
distance_eq1_load5_diame3,    deceleration_eq1_load5_diame3,    velocity1,
time1, pressure1, Fc1, Fb1, nul, Fba1, Fba2, Fba3, Fba4, Fba5, Fba6, Fba7,
Fba8, Fba9, Fdyn1, Fdyn2, Fdyn3, Fdyn4, Fdyn5, Fdyn6, Fdyn7, Fdyn8, Fdyn9,
Fbrake1, Fra1, Fra2, Fra3, Fra4, Fra5, Fra6, Fra7, Fra8, Fra9, Fg1, Fg2,
Fg3, Fg4, Fg5, Fg6, Fg7, Fg8, Fg9, Fext1, FB1, Ftotal1, acc1, s1,
time_eq1, distance_eq1, deceleration_eq1] = load_diam(stp, x, condition,
sav, GeneralitiesVehicleComp_Data, NominalMass_Data, DynamicMass_Data,
WheelDiameter_Data, VehicleResistance_Data, FrictionBrakeForces_Data,
DynamicBrakeForces_Data, TrackBrake_Data, Initial_Data, Degraded_Data_1,
Pressure_Vel_Data);

vectors_zero;
D = {'D1'; 'D2'; 'D3'; 'D4'; 'D5'; 'D6'; 'D7'; 'D8'; 'D9'}; %
Designations.
Diameter = ["New" "Half" "Worn"];

% The number of calculations depends on input load and wheel diameter.
%-----%
load = char(table2array(Degraded_Data_1('Load', 'D1'))); % Input load.
Dim = char(table2array(Degraded_Data_1('Wheel Diameter', 'D1'))); %
Diameter we want to calculate.
if load == "ALL"
    for loads=1:5
        % Brake Disc Diameter
        if Dim == "ALL"
            for diame=1:3
                forces_calculation; % Algorithm that calculates forces.
            end
        end
    end
end

```

```

        end
    else
        for diame=1:3
            if (Dim == Diameter(diame))
                forces_calculation; % Algorithm that calculates forces.
            end
        end
    end
end

end
else
    for loads=1:5
        if (load == char(NominalMass_Data.Properties.RowNames(load)))
            % Brake Disc Diameter
            if Dim == "ALL"
                for diame=1:3
                    forces_calculation; % Algorithm that calculates forces.
                end
            else
                for diame=1:3
                    if (Dim == Diameter(diame))
                        forces_calculation; % Algorithm that calculates
forces.
                    end
                end
            end
        end
    end
end
end
end
end

```

SCRIPT 4. FORCES_CALCULATION

```

% Initialize Vectors
%-----%
n = ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q'
'R' 'S' 'T']; % Different pressure rows. Excel file "Input Data.xlsx"
sheet "5.4".
frict_coeff = {'Friction Coeff 1'; 'Friction Coeff 2'; 'Friction Coeff
3'; 'Friction Coeff 4'; 'Friction Coeff 5'}; % Different friction
coefficients. Excel file "Input Data.xlsx" sheet "5.4".
time = []; % Time vector. Will store every iteration.
s = []; % Distance vector. Will store every iteration.
s_eq = []; % Equivalent distance vector. Will store every iteration.
time(1) = 0; % Initialize first time element to 0.
s(1) = 0; % Initialize first distance element to 0.
s_eq(1) = 0; % Initialize first equivalent distance element to 0.
pressure = []; % Pressure vector. Will store every iteration.
Fba = []; % Braking force per axle vector vector. Will store every
iteration.

```

```
Fba1 = [];  
Fba2 = [];  
Fba3 = [];  
Fba4 = [];  
Fba5 = [];  
Fba6 = [];  
Fba7 = [];  
Fba8 = [];  
Fba9 = [];  
Fba_eq = 0;  
Fdyn = []; % Dynamic force vector. Will store every iteration.  
Fdyn1 = [];  
Fdyn2 = [];  
Fdyn3 = [];  
Fdyn4 = [];  
Fdyn5 = [];  
Fdyn6 = [];  
Fdyn7 = [];  
Fdyn8 = [];  
Fdyn9 = [];  
Fbrake(1) = 0; % Track brake force vector. Will store every iteration.  
Fbrake_design = [];  
Fbrake_design(1) = 0;  
Fext = 0; % External force vector. Will store every iteration.  
FB = []; % Braking force vector. Will store every iteration.  
Ftotal = []; % Total force vector. Will store every iteration.  
acc = []; % Acceleration vector. Will store every iteration.  
Fg = [];  
Fg1 = [];  
Fg2 = [];  
Fg3 = [];  
Fg4 = [];  
Fg5 = [];  
Fg6 = [];  
Fg7 = [];  
Fg8 = [];  
Fg9 = [];  
Fra = [];  
Fra1 = [];  
Fra2 = [];  
Fra3 = [];  
Fra4 = [];  
Fra5 = [];  
Fra6 = [];  
Fra7 = [];  
Fra8 = [];  
Fra9 = [];  
Fc_designation = [];  
  
st_mass = [];  
dyn_mass = [];  
Diam = [];  
brake_type = {};  
Ac = [];  
rigg_eff = [];  
cylinder_eff = [];
```



```

Fr1 = [];
Fr2 = [];
ic = [];
n_dyn = [];
n_frict = [];
rs = [];
ta = [];
tab = [];
teq = [];
trk_type = {};
fr_co = {};
A = [];
B = [];
C = [];
slope = [];
dyn_type = {};
press = {};
F = [];
v1 = [];
v2 = [];

% Initial Velocity
velocity = []; % Velocity vector. Will store every iteration.
velocity_eq = []; % Equivalent velocity vector. Will store every
iteration.
velocity(1) = x;
velocity_eq(1) = x;

% Calculation error.
error = table2array(Initial_Data(1, 'Error'));

% Forces Calculation
%-----%
pass = 0;
i = 1;

for design=1:length(D)

    % Static Mass
    st_mass(design) = table2array(NominalMass_Data(loads,
D(design)));

    if st_mass(design) == 0 % We just do the calculations if static
mass is not equal to zero.
        continue
    end

    pass = 1;

    % Dynamic Mass
    dyn_mass1 = table2array(NominalMass_Data('Rotating Mass, mjrot',
D(design)));
    dyn_mass(design) = dyn_mass1 + st_mass(design);

    % Diameter value.
    Diam(design) = table2array(WheelDiameter_Data(diame, D(design)));

```

```

    % Friction Type of Brake.
    brake_type(design) = table2cell(Degraded_Data_1('Friction Type of
Brake', D(design)));

    % Dynamic Type of Brake.
    dyn_type(design) = table2cell(Degraded_Data_1('Dynamic Type of
Brake', D(design)));

    % Track Type of Brake.
    trk_type(design) = table2cell(Degraded_Data_1('Track Type of
Brake', D(design)));

    % Piston Effective Section.
    Ac(design) = table2array(FrictionBrakeForces_Data('Piston
Effective Section', string(brake_type(design))));

    % Rigging Efficiency.
    rigg_eff(design) = table2array(FrictionBrakeForces_Data('Rigging
Efficiency', string(brake_type(design))));

    % Cylinder Efficiency.
    cylinder_eff(design) =
table2array(FrictionBrakeForces_Data('Cylinder Efficiency',
string(brake_type(design))));

    % Call Spring Force.
    Fr1(design) = table2array(FrictionBrakeForces_Data('Call Spring
Force', string(brake_type(design))));

    % Opposite Regulator Force.
    Fr2(design) = table2array(FrictionBrakeForces_Data('Opposite
Regulator Force', string(brake_type(design))));

    % Wheelhouse Ratio.
    ic(design) = table2array(FrictionBrakeForces_Data('Rigging
Ratio', string(brake_type(design))));

    % Available Friction Brakes.
    n_frict(design) =
str2num(char(table2array(Degraded_Data_1('Available Friction Brakes',
D(design)))));

    % Available Dynamic Brakes.
    n_dynamic(design) =
str2num(char(table2array(Degraded_Data_1('Available Dynamic Brake',
D(design)))));

    % Available Dynamic Brakes.
    eff_dynamic(design) = str2num(char(table2array(Degraded_Data_1('%
Dynamic Brake', D(design)))));

    % Available Dynamic Brakes.
    n_track(design) =
str2num(char(table2array(Degraded_Data_1('Available Track Brake',
D(design)))));

```

```

    % Friction Radius.
    rs(design) = table2array(FrictionBrakeForces_Data('Friction
Radius', string(brake_type(design))));

    % Delay Time.
    ta(design) = table2array(FrictionBrakeForces_Data('Delay Time',
string(brake_type(design))));

    % Delay Time Dyn.
    ta_dyn(design) = table2array(DynamicBrakeForces_Data('Delay
Time', string(dyn_type(design))));

    % Delay Time Track.
    ta_trk(design) = table2array(TrackBrake_Data(9,
string(trk_type(design))));

    % Accumulation Time.
    tab(design) = table2array(FrictionBrakeForces_Data('Accumulation
Time', string(brake_type(design))));

    % Accumulation Time Dyn.
    tab_dyn(design) =
table2array(DynamicBrakeForces_Data('Accumulation
Time',
string(dyn_type(design))));

    % Accumulation Time Track.
    tab_trk(design) = table2array(TrackBrake_Data(10,
string(trk_type(design))));

    % Response Equivalent Time.
    teq(design) = table2array(FrictionBrakeForces_Data('Response
Equivalent Time', string(brake_type(design))));

    % Response Equivalent Time.
    teq_dyn(design) = table2array(DynamicBrakeForces_Data('Response
Equivalent Time', string(dyn_type(design))));

    % Response Equivalent Time.
    teq_trk(design) = table2array(TrackBrake_Data(11,
string(trk_type(design))));

    % Friction Coeff.
    fr_co(design) = table2cell(Degraded_Data_1('Adhesion
Coefficient', D(design)));

    % Pressure.
    press(design) =
table2cell(Degraded_Data_1('Pressure', D(design)));

    % Resistance.
    A(design) = table2array(VehicleResistance_Data('A', 'D1'));
    B(design) = table2array(VehicleResistance_Data('B', 'D1'));
    C(design) = table2array(VehicleResistance_Data('C', 'D1'));
    slope(design) = str2num(char(table2array(Degraded_Data_1('Slope
[mm/m]', D(design))))); % Track slope.

```

```

    % Dynamic Brake Forces.
    F(design) = table2array(DynamicBrakeForces_Data('F',
string(dyn_type(design)))); % Brake force.
    v1(design) = table2array(DynamicBrakeForces_Data('V1',
string(dyn_type(design)))); % Velocity 1
    v2(design) = table2array(DynamicBrakeForces_Data('V2',
string(dyn_type(design)))); % Velocity 2
    %-----%
end

if pass == 1

    while velocity(i)>0 & velocity(i)>error % Calculate while velocity is
greater than 0.

        Fba(i) = 0;
        Fba_eq = 0;
        Fdyn(i) = 0;
        Fbrake(i) = 0;
        Fext = 0;
        Fc(i) = 0;
        Fra(i) = 0;

        for design=1:length(D)

            if st_mass(design) == 0 % We just do the calculations if
static mass is not equal to zero.
                continue
            end

            if n_frict(design) ~= 0
                % Friction Forces calculation.
                %-----%
                % Read the pressure variable, depends on velocity.
                if velocity(i) >=
table2array(Pressure_Vel_Data(string(press(design)),2))
                    pressure(i) =
table2array(Pressure_Vel_Data(string(press(design)),1));
                end

                if velocity(i) >=
table2array(Pressure_Vel_Data(string(press(design)),4)) & velocity(i) <
table2array(Pressure_Vel_Data(string(press(design)),2))
                    pressure(i) =
table2array(Pressure_Vel_Data(string(press(design)),3));
                end

                if velocity(i) >=
table2array(Pressure_Vel_Data(string(press(design)),6)) & velocity(i) <
table2array(Pressure_Vel_Data(string(press(design)),4))
                    pressure(i) =
table2array(Pressure_Vel_Data(string(press(design)),5));
                end

                if velocity(i) >=
table2array(Pressure_Vel_Data(string(press(design)),8)) & velocity(i) <
table2array(Pressure_Vel_Data(string(press(design)),6))

```

```

                                pressure(i) =
table2array(Pressure_Vel_Data(string(press(design)),7));
    end

    % Calculate Cylinder Force.
    Fcc = (10 * pressure(i) * Ac(design) *
cylinder_eff(design)) - Fr1(design) - Fr2(design);
    if Fcc < 0
        Fcc = 0;
    end
    if Fc(i) ~= 0
        if time(i) < str2num(string(ta(design)))
            Fc(i) = Fc_1 + 0;
        end
        if time(i) >= str2num(string(ta(design))) & time(i)
< str2num(string(tab(design)))
            Fc(i) = Fc_1 + (Fcc*time(i)/(tab(design) -
ta(design))) - (Fcc*ta(design)/(tab(design) - ta(design)));
        end
        if time(i) >= str2num(string(tab(design)))
            Fc(i) = Fc_1 + Fcc;
        end
        Fc_designation(design) = Fc(i) - Fc_1;
        Fc_1 = Fc(i);
    else
        if time(i) < str2num(string(ta(design)))
            Fc_1 = 0;
        end
        if time(i) >= str2num(string(ta(design))) & time(i)
< str2num(string(tab(design)))
            Fc_1 = (Fcc*time(i)/(tab(design) - ta(design)))
- (Fcc*ta(design)/(tab(design) - ta(design)));
        end
        if time(i) >= str2num(string(tab(design)))
            Fc_1 = Fcc;
        end
        Fc_designation(design) = Fc_1;
        Fc(i) = Fc_1;
    end

    % Calculate Friction Brake Application Force
    Fb(i) = (Fc(i) * ic(design) * rigg_eff(design));
    Fb_designation(design) = (Fc_designation(design) *
ic(design) * rigg_eff(design));
    Fb_eq = (Fcc * ic(design) * rigg_eff(design));
    Fb(i) = sum(Fb_designation);

    % Friction Coeff. Depends on velocity.
    for cf=1:length(frict_coeff)
        if string(fr_co(design)) == frict_coeff(cf)
            coeff_i = cf;
        end
    end
    switch coeff_i
        case 1
            coeff = 'FrictionCoeff1';

```

```

        case 2
            coeff = 'FrictionCoeff2';
        case 3
            coeff = 'FrictionCoeff3';
        case 4
            coeff = 'FrictionCoeff4';
        case 5
            coeff = 'FrictionCoeff5';
    end

    stop = 0;
    while stop == 0
        for e=1:height(Initial_Data(:, 'Speed'))
            if velocity(i) >
table2array(Initial_Data(e, 'Speed'))
                nu(i) = table2array(Initial_Data(e-1, coeff));
                stop = 1;
            end
        end
    end

    % Calculate Friction Brake Application Force.
    % Fba(i) = Fb(i) * nu(i) * (rs(design)/(Diam(design)/2))
* n_fric(design);
        Fba_eq = Fba_eq + Fb_eq * nu(i) *
(rs(design)/(Diam(design)/2)) * n_fric(design);
        Fba_designation(design) = Fb_designation(design) * nu(i)
* (rs(design)/(Diam(design)/2)) * n_fric(design);
        Fba(i) = sum(Fba_designation);

    if design == 1
        Fba1(i) = Fba_designation(design);
    elseif design == 2
        Fba2(i) = Fba_designation(design);
    elseif design == 3
        Fba3(i) = Fba_designation(design);
    elseif design == 4
        Fba4(i) = Fba_designation(design);
    elseif design == 5
        Fba5(i) = Fba_designation(design);
    elseif design == 6
        Fba6(i) = Fba_designation(design);
    elseif design == 7
        Fba7(i) = Fba_designation(design);
    elseif design == 8
        Fba8(i) = Fba_designation(design);
    elseif design == 9
        Fba9(i) = Fba_designation(design);
    end
end
if n_dynamic(design) ~= 0
    % Dynamic Brake Forces.
    %-----%
    if Fdyn(i) ~= 0
        if velocity(i) < v2(design)
            Fdyn(i) = Fdyn_1 + F(design);
        end
    end
end

```

```

        end
        if velocity(i)>v2(design) & velocity(i)<v1(design)
            Fdyn(i) = Fdyn_1 +
F(design)*v2(design)/velocity(i);
        end
        if velocity(i)>=v1(design)
            Fdyn(i) = Fdyn_1 +
F(design)*v2(design)*v1(design)/(velocity(i))^2;
        end
        Fdyn_designation = Fdyn(i)-Fdyn_1;
        if time(i) < str2num(string(ta_dyn(design)))
            Fdyn(i) = 0;
            Fdyn_designation = 0;
        end
        if time(i) >= str2num(string(ta_dyn(design))) &
time(i) < str2num(string(tab_dyn(design)))
            Fdyn(i) = (Fdyn(i)*time(i)/(tab_dyn(design) -
ta_dyn(design))) -
(Fdyn(i)*ta_dyn(design)/(tab_dyn(design) -
ta_dyn(design)));
            Fdyn_designation =
(Fdyn_designation*time(i)/(tab_dyn(design) -
ta_dyn(design))) -
(Fdyn_designation*ta_dyn(design)/(tab_dyn(design) -
ta_dyn(design)));
        end
        if time(i) >= str2num(string(tab_dyn(design)))
            Fdyn(i) = Fdyn(i);
            Fdyn_designation = Fdyn_designation;
        end
    else
        if velocity(i)<=v2(design)
            Fdyn_1 = F(design);
        end
        if velocity(i)>v2(design) & velocity(i)<v1(design)
            Fdyn_1 = F(design)*v2(design)/velocity(i);
        end
        if velocity(i)>=v1(design)
            Fdyn_1 =
F(design)*v2(design)*v1(design)/((velocity(i))^2);
        end
        Fdyn_designation = Fdyn_1;
        if time(i) < str2num(string(ta_dyn(design)))
            Fdyn_1 = 0;
            Fdyn_designation = 0;
        end
        if time(i) >= str2num(string(ta_dyn(design))) &
time(i) < str2num(string(tab_dyn(design)))
            Fdyn_1 = (Fdyn_1*time(i)/(tab_dyn(design) -
ta_dyn(design))) -
(Fdyn_1*ta_dyn(design)/(tab_dyn(design) -
ta_dyn(design)));
            Fdyn_designation =
(Fdyn_designation*time(i)/(tab_dyn(design) -
ta_dyn(design))) -
(Fdyn_designation*ta_dyn(design)/(tab_dyn(design) -
ta_dyn(design)));
        end
        if time(i) >= str2num(string(tab_dyn(design)))
            Fdyn_1 = Fdyn_1;
            Fdyn_designation = Fdyn_designation;
        end
    end
end

```

```

        Fdyn(i) = Fdyn_1;
    end
    Fdyn(i) = eff_dynamic(design)*Fdyn(i)*n_dynamic(design);
    if design == 1
        Fdyn1(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 2
        Fdyn2(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 3
        Fdyn3(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 4
        Fdyn4(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 5
        Fdyn5(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 6
        Fdyn6(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 7
        Fdyn7(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 8
        Fdyn8(i) = eff_dynamic(design)*(Fdyn_designation);
    elseif design == 9
        Fdyn9(i) = eff_dynamic(design)*(Fdyn_designation);
    end
end
if n_track(design) ~= 0
    % Track Brake Forces.
    %-----%
    stop = 0;
    while stop == 0
        for e=1:8
            if velocity(i) >
str2num(string(table2array(TrackBrake_Data(e, 'Speed'))))
                Fbrake_design(design) =
str2num(string(table2array(TrackBrake_Data(e-
1,string(trk_type(design))))));
                stop = 1;
                break;
            end
        end
    end
    Fbrake_design(design) = Fbrake_design(design) *
n_track(design);
    Fbrake(i) = Fbrake(i) + Fbrake_design(design);
end
if time(i) < str2num(string(ta_trk(design)))
    Fbrake(i) = 0;
end
if time(i) >= str2num(string(ta_trk(design))) & time(i) <
str2num(string(tab_trk(design)))
    Fbrake(i) = (Fbrake(i)*time(i)/(tab_trk(design) -
ta_trk(design))) - (Fbrake(i)*ta_trk(design)/(tab_trk(design) -
ta_trk(design)));
end
if time(i) >= str2num(string(tab_trk(design)))
    Fbrake(i) = Fbrake(i);
end
end

```



```

% External Forces.
%-----%
if Fra(i) ~= 0
    Fra(i) = Fra_1 + A(design) + B(design) * velocity(i) +
C(design) * (velocity(i))^2;
    Fg(i) = Fg_1 + st_mass(design) * 9.81 * slope(design)/1000;
    Fext(i) = Fra(i)+Fg(i)*1000;
    Fra_designation = Fra(i) - Fra_1;
    Fg_designation = Fg(i) - Fg_1;
else
    Fra_1 = A(design) + B(design) * velocity(i) + C(design)
* (velocity(i))^2;
    Fra(i) = Fra_1;
    Fg_1 = st_mass(design) * 9.81 * slope(design)/1000;
    Fg(i) = Fg_1;
    Fext(i) = Fra_1+Fg_1*1000;
    Fra_designation = Fra(i);
    Fg_designation = Fg(i);
end
if design == 1
    Fra1(i) = Fra_designation;
    Fg1(i) = Fg_designation;
elseif design == 2
    Fra2(i) = Fra_designation;
    Fg2(i) = Fg_designation;
elseif design == 3
    Fra3(i) = Fra_designation;
    Fg3(i) = Fg_designation;
elseif design == 4
    Fra4(i) = Fra_designation;
    Fg4(i) = Fg_designation;
elseif design == 5
    Fra5(i) = Fra_designation;
    Fg5(i) = Fg_designation;
elseif design == 6
    Fra6(i) = Fra_designation;
    Fg6(i) = Fg_designation;
elseif design == 7
    Fra7(i) = Fra_designation;
    Fg7(i) = Fg_designation;
elseif design == 8
    Fra8(i) = Fra_designation;
    Fg8(i) = Fg_designation;
elseif design == 9
    Fra9(i) = Fra_designation;
    Fg9(i) = Fg_designation;
end

end

if pass == 0
    break
end

% Sum Brakes Forces.
%-----%

```

```

FB(i) = Fba(i) + Fdyn(i) + Fbrake(i);
FB_eq = Fba_eq + Fdyn(i) + Fbrake(i);

% Acceleration Calculation.
%-----%
dyn_mass_t = sum(dyn_mass);
Ftotal(i) = FB(i) + Fext(i);
Ftotal_eq = FB_eq/1000 + Fext(i);
acc(i) = Ftotal(i) / (dyn_mass_t * 1000);
acc_eq(i) = Ftotal_eq / dyn_mass_t;

% Iterate.
%-----%
time(i+1) = time(i) + stp;
velocity(i+1) = velocity(i) - acc(i) * stp * 3.6;
s(i+1) = s(i) + 0.27778*velocity(i)*stp - (1/2)*acc(i)*stp*stp;

i = i+1;

% Sf(t) = 100%
%-----%
if velocity_eq(end) > 0 & velocity(end)>error
    velocity_eq(i) = velocity_eq(i-1) - acc_eq(i-1) * stp * 3.6;
    s_eq(i) = s_eq(i-1) + 0.27778*velocity_eq(i-1)*stp -
(1/2)*acc_eq(i-1)*stp*stp;
end
end

% Equivalent calculations.
%-----%
velocity_eq(end) = [];
s_eq(end) = [];
time(end) = [];
velocity(end) = [];
s(end) = [];

vel_1 = velocity(1)/3.6;
time_eq = (s(end)-s_eq(end))/ vel_1;
distance_eq = vel_1*time_eq;
vel_2 = velocity(end)/3.6;
deceleration_eq = ((vel_1^2)-(vel_2^2))/(2*s_eq(end));

% Save calculation in corresponding vectors.
%-----%
vectors;
if sav == 1
    equal_len;
    if condition == 1
        export_cond1_txt;
    elseif condition == 2
        export_cond2_txt;
    elseif condition == 3
        export_cond3_txt;
    elseif condition == 4
        export_cond4_txt;
    end
end

```

```
elseif condition == 5
    export_cond5_txt;
end
end
end
```

SCRIPT 5. VECTORS

```
if loads == 1 & diame == 1
    velocity1_load1_diame1 = velocity; % Saves output data in detailed
vector. % Saves output data in detailed vector.
    time1_load1_diame1 = time; % Saves output data in detailed vector.
    pressure1_load1_diame1 = pressure; % Saves output data in detailed
vector.
    Fc1_load1_diame1 = Fc; % Saves output data in detailed vector.
    Fb1_load1_diame1 = Fb; % Saves output data in detailed vector.
    nu1_load1_diame1 = nu; % Saves output data in detailed vector.
    Fba1_load1_diame1 = Fba; % Saves output data in detailed vector.
    Fdyn1_load1_diame1 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load1_diame1 = Fbrake; % Saves output data in detailed vector.
    Fra1_load1_diame1 = Fra; % Saves output data in detailed vector.
    Fg1_load1_diame1 = Fg; % Saves output data in detailed vector.
    Fext1_load1_diame1 = Fext; % Saves output data in detailed vector.
    FB1_load1_diame1 = FB; % Saves output data in detailed vector.
    Ftotal1_load1_diame1 = Ftotal; % Saves output data in detailed vector.
    acc1_load1_diame1 = acc; % Saves output data in detailed vector.
    s1_load1_diame1 = s; % Saves output data in detailed vector.
    time_eq1_load1_diame1 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load1_diame1 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load1_diame1 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 1 & diame == 2
    velocity1_load1_diame2 = velocity; % Saves output data in detailed
vector.
    time1_load1_diame2 = time; % Saves output data in detailed vector.
    pressure1_load1_diame2 = pressure; % Saves output data in detailed
vector.
    Fc1_load1_diame2 = Fc; % Saves output data in detailed vector.
    Fb1_load1_diame2 = Fb; % Saves output data in detailed vector.
    nu1_load1_diame2 = nu; % Saves output data in detailed vector.
    Fba1_load1_diame2 = Fba; % Saves output data in detailed vector.
    Fdyn1_load1_diame2 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load1_diame2 = Fbrake; % Saves output data in detailed vector.
    Fra1_load1_diame2 = Fra; % Saves output data in detailed vector.
    Fg1_load1_diame2 = Fg; % Saves output data in detailed vector.
    Fext1_load1_diame2 = Fext; % Saves output data in detailed vector.
    FB1_load1_diame2 = FB; % Saves output data in detailed vector.
    Ftotal1_load1_diame2 = Ftotal; % Saves output data in detailed vector.
    acc1_load1_diame2 = acc; % Saves output data in detailed vector.
    s1_load1_diame2 = s; % Saves output data in detailed vector.
    time_eq1_load1_diame2 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load1_diame2 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load1_diame2 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 1 & diame == 3
```

```
velocity1_load1_diame3 = velocity; % Saves output data in detailed
vector.
time1_load1_diame3 = time; % Saves output data in detailed vector.
pressure1_load1_diame3 = pressure; % Saves output data in detailed
vector.
Fc1_load1_diame3 = Fc; % Saves output data in detailed vector.
Fb1_load1_diame3 = Fb; % Saves output data in detailed vector.
nu1_load1_diame3 = nu; % Saves output data in detailed vector.
Fba1_load1_diame3 = Fba; % Saves output data in detailed vector.
Fdyn1_load1_diame3 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load1_diame3 = Fbrake; % Saves output data in detailed vector.
Fra1_load1_diame3 = Fra; % Saves output data in detailed vector.
Fg1_load1_diame3 = Fg; % Saves output data in detailed vector.
Fext1_load1_diame3 = Fext; % Saves output data in detailed vector.
FB1_load1_diame3 = FB; % Saves output data in detailed vector.
Ftotal1_load1_diame3 = Ftotal; % Saves output data in detailed vector.
acc1_load1_diame3 = acc; % Saves output data in detailed vector.
s1_load1_diame3 = s; % Saves output data in detailed vector.
time_eq1_load1_diame3 = time_eq; % Saves output data in detailed
vector.
distance_eq1_load1_diame3 = distance_eq; % Saves output data in
detailed vector.
deceleration_eq1_load1_diame3 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 2 & diame == 1
velocity1_load2_diame1 = velocity; % Saves output data in detailed
vector.
time1_load2_diame1 = time; % Saves output data in detailed vector.
pressure1_load2_diame1 = pressure; % Saves output data in detailed
vector.
Fc1_load2_diame1 = Fc; % Saves output data in detailed vector.
Fb1_load2_diame1 = Fb; % Saves output data in detailed vector.
nu1_load2_diame1 = nu; % Saves output data in detailed vector.
Fba1_load2_diame1 = Fba; % Saves output data in detailed vector.
Fdyn1_load2_diame1 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load2_diame1 = Fbrake; % Saves output data in detailed vector.
Fra1_load2_diame1 = Fra; % Saves output data in detailed vector.
Fg1_load2_diame1 = Fg; % Saves output data in detailed vector.
Fext1_load2_diame1 = Fext; % Saves output data in detailed vector.
FB1_load2_diame1 = FB; % Saves output data in detailed vector.
Ftotal1_load2_diame1 = Ftotal; % Saves output data in detailed vector.
acc1_load2_diame1 = acc; % Saves output data in detailed vector.
s1_load2_diame1 = s; % Saves output data in detailed vector.
time_eq1_load2_diame1 = time_eq; % Saves output data in detailed
vector.
distance_eq1_load1_diame1 = distance_eq; % Saves output data in
detailed vector.
deceleration_eq1_load2_diame1 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 2 & diame == 2
velocity1_load2_diame2 = velocity; % Saves output data in detailed
vector.
time1_load2_diame2 = time; % Saves output data in detailed vector.
pressure1_load2_diame2 = pressure; % Saves output data in detailed
vector.
Fc1_load2_diame2 = Fc; % Saves output data in detailed vector.
```

```
Fb1_load2_diame2 = Fb; % Saves output data in detailed vector.
nul_load2_diame2 = nu; % Saves output data in detailed vector.
Fba1_load2_diame2 = Fba; % Saves output data in detailed vector.
Fdyn1_load2_diame2 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load2_diame2 = Fbrake; % Saves output data in detailed vector.
Fra1_load2_diame2 = Fra; % Saves output data in detailed vector.
Fg1_load2_diame2 = Fg; % Saves output data in detailed vector.
Fext1_load2_diame2 = Fext; % Saves output data in detailed vector.
FB1_load2_diame2 = FB; % Saves output data in detailed vector.
Ftotal1_load2_diame2 = Ftotal; % Saves output data in detailed vector.
acc1_load2_diame2 = acc; % Saves output data in detailed vector.
s1_load2_diame2 = s; % Saves output data in detailed vector.
time_eq1_load2_diame2 = time_eq; % Saves output data in detailed
vector.
distance_eq1_load1_diame2 = distance_eq; % Saves output data in
detailed vector.
deceleration_eq1_load2_diame2 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 2 & diame == 3
velocity1_load2_diame3 = velocity; % Saves output data in detailed
vector.
time1_load2_diame3 = time; % Saves output data in detailed vector.
pressure1_load2_diame3 = pressure; % Saves output data in detailed
vector.
Fc1_load2_diame3 = Fc; % Saves output data in detailed vector.
Fb1_load2_diame3 = Fb; % Saves output data in detailed vector.
nul_load2_diame3 = nu; % Saves output data in detailed vector.
Fba1_load2_diame3 = Fba; % Saves output data in detailed vector.
Fdyn1_load2_diame3 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load2_diame3 = Fbrake; % Saves output data in detailed vector.
Fra1_load2_diame3 = Fra; % Saves output data in detailed vector.
Fg1_load2_diame3 = Fg; % Saves output data in detailed vector.
Fext1_load2_diame3 = Fext; % Saves output data in detailed vector.
FB1_load2_diame3 = FB; % Saves output data in detailed vector.
Ftotal1_load2_diame3 = Ftotal; % Saves output data in detailed vector.
acc1_load2_diame3 = acc; % Saves output data in detailed vector.
s1_load2_diame3 = s; % Saves output data in detailed vector.
time_eq1_load2_diame3 = time_eq; % Saves output data in detailed
vector.
distance_eq1_load1_diame3 = distance_eq; % Saves output data in
detailed vector.
deceleration_eq1_load2_diame3 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 3 & diame == 1
velocity1_load3_diamel = velocity; % Saves output data in detailed
vector.
time1_load3_diamel = time; % Saves output data in detailed vector.
pressure1_load3_diamel = pressure; % Saves output data in detailed
vector.
Fc1_load3_diamel = Fc; % Saves output data in detailed vector.
Fb1_load3_diamel = Fb; % Saves output data in detailed vector.
nul_load3_diamel = nu; % Saves output data in detailed vector.
Fba1_load3_diamel = Fba; % Saves output data in detailed vector.
Fdyn1_load3_diamel = Fdyn; % Saves output data in detailed vector.
Fbrake1_load3_diamel = Fbrake; % Saves output data in detailed vector.
Fra1_load3_diamel = Fra; % Saves output data in detailed vector.
```

```
Fg1_load3_diame1 = Fg; % Saves output data in detailed vector.
Fext1_load3_diame1 = Fext; % Saves output data in detailed vector.
FB1_load3_diame1 = FB; % Saves output data in detailed vector.
Ftotal1_load3_diame1 = Ftotal; % Saves output data in detailed vector.
acc1_load3_diame1 = acc; % Saves output data in detailed vector.
s1_load3_diame1 = s; % Saves output data in detailed vector.
time_eq1_load3_diame1 = time_eq; % Saves output data in detailed
vector.
distance_eq1_load3_diame1 = distance_eq; % Saves output data in
detailed vector.
deceleration_eq1_load3_diame1 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 3 & diame == 2
velocity1_load3_diame2 = velocity; % Saves output data in detailed
vector.
time1_load3_diame2 = time; % Saves output data in detailed vector.
pressure1_load3_diame2 = pressure; % Saves output data in detailed
vector.
Fc1_load3_diame2 = Fc; % Saves output data in detailed vector.
Fb1_load3_diame2 = Fb; % Saves output data in detailed vector.
nul_load3_diame2 = nu; % Saves output data in detailed vector.
Fba1_load3_diame2 = Fba; % Saves output data in detailed vector.
Fdyn1_load3_diame2 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load3_diame2 = Fbrake; % Saves output data in detailed vector.
Fra1_load3_diame2 = Fra; % Saves output data in detailed vector.
Fg1_load3_diame2 = Fg; % Saves output data in detailed vector.
Fext1_load3_diame2 = Fext; % Saves output data in detailed vector.
FB1_load3_diame2 = FB; % Saves output data in detailed vector.
Ftotal1_load3_diame2 = Ftotal; % Saves output data in detailed vector.
acc1_load3_diame2 = acc; % Saves output data in detailed vector.
s1_load3_diame2 = s; % Saves output data in detailed vector.
time_eq1_load3_diame2 = time_eq; % Saves output data in detailed
vector.
distance_eq1_load3_diame2 = distance_eq; % Saves output data in
detailed vector.
deceleration_eq1_load3_diame2 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 3 & diame == 3
velocity1_load3_diame3 = velocity; % Saves output data in detailed
vector.
time1_load3_diame3 = time; % Saves output data in detailed vector.
pressure1_load3_diame3 = pressure; % Saves output data in detailed
vector.
Fc1_load3_diame3 = Fc; % Saves output data in detailed vector.
Fb1_load3_diame3 = Fb; % Saves output data in detailed vector.
nul_load3_diame3 = nu; % Saves output data in detailed vector.
Fba1_load3_diame3 = Fba; % Saves output data in detailed vector.
Fdyn1_load3_diame3 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load3_diame3 = Fbrake; % Saves output data in detailed vector.
Fra1_load3_diame3 = Fra; % Saves output data in detailed vector.
Fg1_load3_diame3 = Fg; % Saves output data in detailed vector.
Fext1_load3_diame3 = Fext; % Saves output data in detailed vector.
FB1_load3_diame3 = FB; % Saves output data in detailed vector.
Ftotal1_load3_diame3 = Ftotal; % Saves output data in detailed vector.
acc1_load3_diame3 = acc; % Saves output data in detailed vector.
s1_load3_diame3 = s; % Saves output data in detailed vector.
```

```
    time_eq1_load3_diame3 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load3_diame3 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load3_diame3 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 4 & diame == 1
    velocity1_load4_diame1 = velocity; % Saves output data in detailed
vector.
    time1_load4_diame1 = time; % Saves output data in detailed vector.
    pressure1_load4_diame1 = pressure; % Saves output data in detailed
vector.
    Fc1_load4_diame1 = Fc; % Saves output data in detailed vector.
    Fb1_load4_diame1 = Fb; % Saves output data in detailed vector.
    nu1_load4_diame1 = nu; % Saves output data in detailed vector.
    Fba1_load4_diame1 = Fba; % Saves output data in detailed vector.
    Fdyn1_load4_diame1 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load4_diame1 = Fbrake; % Saves output data in detailed vector.
    Fra1_load4_diame1 = Fra; % Saves output data in detailed vector.
    Fg1_load4_diame1 = Fg; % Saves output data in detailed vector.
    Fext1_load4_diame1 = Fext; % Saves output data in detailed vector.
    FB1_load4_diame1 = FB; % Saves output data in detailed vector.
    Ftotal1_load4_diame1 = Ftotal; % Saves output data in detailed vector.
    acc1_load4_diame1 = acc; % Saves output data in detailed vector.
    s1_load4_diame1 = s; % Saves output data in detailed vector.
    time_eq1_load4_diame1 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load4_diame1 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load4_diame1 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 4 & diame == 2
    velocity1_load4_diame2 = velocity; % Saves output data in detailed
vector.
    time1_load4_diame2 = time; % Saves output data in detailed vector.
    pressure1_load4_diame2 = pressure; % Saves output data in detailed
vector.
    Fc1_load4_diame2 = Fc; % Saves output data in detailed vector.
    Fb1_load4_diame2 = Fb; % Saves output data in detailed vector.
    nu1_load4_diame2 = nu; % Saves output data in detailed vector.
    Fba1_load4_diame2 = Fba; % Saves output data in detailed vector.
    Fdyn1_load4_diame2 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load4_diame2 = Fbrake; % Saves output data in detailed vector.
    Fra1_load4_diame2 = Fra; % Saves output data in detailed vector.
    Fg1_load4_diame2 = Fg; % Saves output data in detailed vector.
    Fext1_load4_diame2 = Fext; % Saves output data in detailed vector.
    FB1_load4_diame2 = FB; % Saves output data in detailed vector.
    Ftotal1_load4_diame2 = Ftotal; % Saves output data in detailed vector.
    acc1_load4_diame2 = acc; % Saves output data in detailed vector.
    s1_load4_diame2 = s; % Saves output data in detailed vector.
    time_eq1_load4_diame2 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load4_diame2 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load4_diame2 = deceleration_eq; % Saves output data
in detailed vector.
```



```
elseif loads == 4 & diame == 3
    velocity1_load4_diame3 = velocity; % Saves output data in detailed
vector.
    time1_load4_diame3 = time; % Saves output data in detailed vector.
    pressure1_load4_diame3 = pressure; % Saves output data in detailed
vector.
    Fc1_load4_diame3 = Fc; % Saves output data in detailed vector.
    Fb1_load4_diame3 = Fb; % Saves output data in detailed vector.
    nul_load4_diame3 = nu; % Saves output data in detailed vector.
    Fba1_load4_diame3 = Fba; % Saves output data in detailed vector.
    Fdyn1_load4_diame3 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load4_diame3 = Fbrake; % Saves output data in detailed vector.
    Fra1_load4_diame3 = Fra; % Saves output data in detailed vector.
    Fg1_load4_diame3 = Fg; % Saves output data in detailed vector.
    Fext1_load4_diame3 = Fext; % Saves output data in detailed vector.
    FB1_load4_diame3 = FB; % Saves output data in detailed vector.
    Ftotal1_load4_diame3 = Ftotal; % Saves output data in detailed vector.
    accl_load4_diame3 = acc; % Saves output data in detailed vector.
    s1_load4_diame3 = s; % Saves output data in detailed vector.
    time_eq1_load4_diame3 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load4_diame3 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load4_diame3 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 5 & diame == 1
    velocity1_load5_diame1 = velocity; % Saves output data in detailed
vector.
    time1_load5_diame1 = time; % Saves output data in detailed vector.
    pressure1_load5_diame1 = pressure; % Saves output data in detailed
vector.
    Fc1_load5_diame1 = Fc; % Saves output data in detailed vector.
    Fb1_load5_diame1 = Fb; % Saves output data in detailed vector.
    nul_load5_diame1 = nu; % Saves output data in detailed vector.
    Fba1_load5_diame1 = Fba; % Saves output data in detailed vector.
    Fdyn1_load5_diame1 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load5_diame1 = Fbrake; % Saves output data in detailed vector.
    Fra1_load5_diame1 = Fra; % Saves output data in detailed vector.
    Fg1_load5_diame1 = Fg; % Saves output data in detailed vector.
    Fext1_load5_diame1 = Fext; % Saves output data in detailed vector.
    FB1_load5_diame1 = FB; % Saves output data in detailed vector.
    Ftotal1_load5_diame1 = Ftotal; % Saves output data in detailed vector.
    accl_load5_diame1 = acc; % Saves output data in detailed vector.
    s1_load5_diame1 = s; % Saves output data in detailed vector.
    time_eq1_load5_diame1 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load5_diame1 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load5_diame1 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 5 & diame == 2
    velocity1_load5_diame2 = velocity; % Saves output data in detailed
vector.
    time1_load5_diame2 = time; % Saves output data in detailed vector.
    pressure1_load5_diame2 = pressure; % Saves output data in detailed
vector.
```

```
Fc1_load5_diame2 = Fc; % Saves output data in detailed vector.
Fb1_load5_diame2 = Fb; % Saves output data in detailed vector.
nul_load5_diame2 = nu; % Saves output data in detailed vector.
Fba1_load5_diame2 = Fba; % Saves output data in detailed vector.
Fdyn1_load5_diame2 = Fdyn; % Saves output data in detailed vector.
Fbrake1_load5_diame2 = Fbrake; % Saves output data in detailed vector.
Fra1_load5_diame2 = Fra; % Saves output data in detailed vector.
Fg1_load5_diame2 = Fg; % Saves output data in detailed vector.
Fext1_load5_diame2 = Fext; % Saves output data in detailed vector.
FB1_load5_diame2 = FB; % Saves output data in detailed vector.
Ftotal1_load5_diame2 = Ftotal; % Saves output data in detailed vector.
acc1_load5_diame2 = acc; % Saves output data in detailed vector.
s1_load5_diame2 = s; % Saves output data in detailed vector.
time_eq1_load5_diame2 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load5_diame2 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load5_diame2 = deceleration_eq; % Saves output data
in detailed vector.
elseif loads == 5 & diame == 3
    velocity1_load5_diame3 = velocity; % Saves output data in detailed
vector.
    time1_load5_diame3 = time; % Saves output data in detailed vector.
    pressure1_load5_diame3 = pressure; % Saves output data in detailed
vector.
    Fc1_load5_diame3 = Fc; % Saves output data in detailed vector.
    Fb1_load5_diame3 = Fb; % Saves output data in detailed vector.
    nul_load5_diame3 = nu; % Saves output data in detailed vector.
    Fba1_load5_diame3 = Fba; % Saves output data in detailed vector.
    Fdyn1_load5_diame3 = Fdyn; % Saves output data in detailed vector.
    Fbrake1_load5_diame3 = Fbrake; % Saves output data in detailed vector.
    Fra1_load5_diame3 = Fra; % Saves output data in detailed vector.
    Fg1_load5_diame3 = Fg; % Saves output data in detailed vector.
    Fext1_load5_diame3 = Fext; % Saves output data in detailed vector.
    FB1_load5_diame3 = FB; % Saves output data in detailed vector.
    Ftotal1_load5_diame3 = Ftotal; % Saves output data in detailed vector.
    acc1_load5_diame3 = acc; % Saves output data in detailed vector.
    s1_load5_diame3 = s; % Saves output data in detailed vector.
    time_eq1_load5_diame3 = time_eq; % Saves output data in detailed
vector.
    distance_eq1_load5_diame3 = distance_eq; % Saves output data in
detailed vector.
    deceleration_eq1_load5_diame3 = deceleration_eq; % Saves output data
in detailed vector.
else
    velocity1 = velocity; % Saves output data in detailed vector.
    time1 = time; % Saves output data in detailed vector.
    pressure1 = pressure; % Saves output data in detailed vector.
    Fc1 = Fc; % Saves output data in detailed vector.
    Fb1 = Fb; % Saves output data in detailed vector.
    nul = nu; % Saves output data in detailed vector.
    Fba1 = Fba; % Saves output data in detailed vector.
    Fdyn1 = Fdyn; % Saves output data in detailed vector.
    Fbrake1 = Fbrake; % Saves output data in detailed vector.
    Fra1 = Fra; % Saves output data in detailed vector.
    Fg1 = Fg; % Saves output data in detailed vector.
```

```
Fext1 = Fext; % Saves output data in detailed vector.  
FB1 = FB; % Saves output data in detailed vector.  
Ftotal1 = Ftotal; % Saves output data in detailed vector.  
acc1 = acc; % Saves output data in detailed vector.  
s1 = s; % Saves output data in detailed vector.  
time_eq1 = time_eq; % Saves output data in detailed vector.  
distance_eq1 = distance_eq; % Saves output data in detailed vector.  
deceleration_eq1 = deceleration_eq; % Saves output data in detailed  
vector.  
end
```



```
), Fba2(:), Fba3(:), Fba4(:), Fba5(:), Fba6(:), Fba7(:), Fba8(:),  
Fba9(:), Fdyn1(:), Fdyn2(:), Fdyn3(:), Fdyn4(:), Fdyn5(:), Fdyn6(:),  
Fdyn7(:), Fdyn8(:), Fdyn9(:), Fra1(:), Fra2(:), Fra3(:), Fra4(:),  
Fra5(:), Fra6(:), Fra7(:), Fra8(:), Fra9(:), Fg1(:), Fg2(:), Fg3(:),  
Fg4(:), Fg5(:), Fg6(:), Fg7(:), Fg8(:), Fg9(:)].');  
fclose(fileID);
```

end

Adicionalmente, los scripts *export_cond2_txt*, *export_cond3_txt*, *export_cond4_txt*, y *export_cond5_txt* mantienen la misma estructura que *export_cond1_txt*, pero almacenando la información en sus respectivas variables.

SCRIPT 7. FILE2EXCEL

```
sample_file = readtable('result_cond1_load1_diam1.txt');  
filename = 'Results.xlsx';  
writetable(sample_file,filename,'Sheet','Data','WriteVariableNames',true,  
 'Range','A1');
```